

附光盘（视频讲解+源代码）

网站开发非常之旅



技术支持：QQ群(21948168)／ 论坛(http://www.rzchina.net)

强锋科技

陈营辉　编著

PHP 网络编程从入门到精通

- 语言通俗，可让初学者快速入门
- 通过数百段典型代码，让您体会PHP的强大功能
- 注重实效，例子只需简单修改即可直接使用
- 讲解了4个典型模块和2个大型案例，以展示项目开发的流程
- 代码规范，层次清楚，注释丰富，易于理解
- 配视频演示光盘，方便学习

清华大学出版社

网站开发非常之旅

PHP 网络编程从入门到精通

强锋科技

陈营辉　编著

清华大学出版社

北　京

内 容 简 介

本书由浅入深地讲解了 PHP 的各项知识。主要包括 PHP 的常量与变量、运算符与表达式、流程控制、函数、数据处理、文件应用、主机信息、图像处理、Session 与 Cookie、正则表达式的使用、面向对象编程、MySQL 数据库知识。另外，本书还精选了多个常用模块，分别是计数器程序、网上投票系统、留言板程序、简单的 BBS 系统、博客程序、网上商城系统。实例结合实际功能需要，符合系统设计要求，并且都是当前流行的应用系统。

本书适合希望从事或正在从事 PHP+MySQL 系统开发的人员阅读、借鉴和参考，同时也适合各类学校作为教材使用。另外，它还是学生毕业设计的好参考书。

图书在版编目（CIP）数据

PHP 网络编程从入门到精通/陈营辉编著. —北京：清华大学出版社，2007.6
（网站开发非常之旅）

ISBN 978-7-302-15412-9

I. P…　II. 陈…　III. PHP 语言-程序设计　IV. TP312

中国版本图书馆 CIP 数据核字（2007）第 084841 号

责任编辑： 欧振旭　邵　兴
封面设计： 范华明
版式设计： 侯哲芬
责任校对： 王文鹏　焦章英
责任印制：
出版发行： 清华大学出版社　　**地　　址：** 北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：** 100084
c-service@tup.tsinghua.edu.cn
社 总 机： 010-62770175　　**邮购热线：** 010-62786544
投稿咨询： 010-62772015　　**客户服务：** 010-62776969
印 刷 者：
装 订 者：
经　　销： 全国新华书店
开　　本： 203×260　**印　　张：** 35　**字　　数：** 776 千字
（附光盘 1 张）
版　　次： 2007 年 6 月第 1 版　**印　　次：** 2007 年 6 月第 1 次印刷
印　　数： 1～6000
定　　价： 59.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：025075-01

编 辑 寄 语

“不积跬步，无以至千里。”

——中国思想家荀子

“网站开发非常之旅”这套书自 2006 年开始已经陆续推出了近 10 个品种，这些品种大都受到了读者的热烈欢迎。自出版后这套书中的大多数图书已经多次印刷，销售居同类图书前列。对于计算机图书，这是一个让人振奋的结果。

根据一年以来读者的反馈可以看出，他们比较一致地认为这套书定位明确，内容有特色，编写质量较好，看后能学到真正有用的东西，而且售后服务和技术支持做得非常好。我们从中不难看出这套书受欢迎的原因。

为了让读者更加全面地了解这套书，下面具体介绍这套书的定位、内容、特色、读者对象等。

丛书定位

根据读者的实际需求，本丛书并不追求面面俱到，而是**定位于实用，并注重对基本知识点的掌握和对基本技能的提升。**突出体现在以下几点：

- ☑ 每本书大体都对应着相应的工作岗位，着重让读者掌握一项技能，使他们在学完一本书以后，可以将所学应用到实际工作中去。
- ☑ 讲解概念，但并不拘泥于概念，而是侧重于对概念的灵活运用，从而让读者在实践中加深对概念的理解和对基本知识的掌握。
- ☑ 不安排纯演示性实例，那种实例通常没有任何应用价值，读者很难通过它而达到较好的应用水平。本丛书中的实例大多选自于实际开发，是对作者多年开发经验的总结。读者通过对这些实例的学习，可以真正体验实际的开发过程，从而将所学应用于实践。

丛书书目

第一批：

《ASP 网络编程从入门到精通》

《ASP.NET 网络数据库开发实例精解》

《ASP+SQL Server 典型网站建设案例》

《服务器配置全攻略》

第二批：

《ASP.NET 2.0 网络编程从入门到精通》

《Dreamweaver 网页制作与色彩搭配全攻略》

《ASP.NET 2.0+SQL Server 网络应用系统开发案例精解》

《JavaScript 网页特效实例大全》

《HTML 网页设计参考手册》

第三批：

《JSP 网络编程从入门到精通》
《PHP 网络编程从入门到精通》
《PHP+MySQL 经典案例剖析》
《Ajax 经典案例开发大全》
《Dreamweaver 网页设计与制作完全手册》
《Dreamweaver+ASP 动态网站开发从入门到精通》
《CSS 标准网页布局开发指南》

丛书特色

☑ 本丛书**按照网站开发的流程组织内容**，从最初的服务器配置，到后台编码的实现，到前台网页设计等内容均有涉及，真正实现了网站开发一条龙。学完本丛书，读者可以全面掌握网站开发的各项技术。
☑ **作者均为有丰富的网络编程经验的一线开发人员**，可以确保技术的先进性、实用性和深入性。
☑ 内容讲解到位，避免空洞，**每个知识点都配有实例**，读者可以上机操作，体会其中的奥妙。
☑ **贯穿丰富的应用实例，并专门设计了综合案例，**让读者真正做到学以致用，并领会项目开发的整体思路。
☑ **配视频讲解光盘，**将一些简单的、操作性强的内容放到光盘中，以加速读者的学习，并节省篇幅，降低定价，让读者真正得到实惠。
☑ **提供必要的售后服务。**从书的每本书都有相应的技术论坛与作者 E-mail（见前言），读者若有疑问，可以按此寻求帮助。

读者对象

☑ 网站建设及网络开发人员
☑ 网页设计及制作人员
☑ 网站维护人员
☑ 网页制作爱好者
☑ 大中专院校的学生
☑ 毕业设计的学生

“不积跬步，无以至千里”。我想这句话可能最能概括这套书成功的真正原因。这套书要想走得远，就得一步一个脚印地做好各个环节的工作。从对读者需求的调研，到确立写作思路，到选择作者，到后期编辑及制作，乃至技术服务等，都得一一落实。只有这样，才能给这套书的继续发展奠定基础，才能让这套书继续成为广大读者的良师益友。

在“网站开发非常之旅”第三批图书即将出版之时，作为这套书的编辑，自是有感而发，希望这套书能一如既往地以高质量、高品质和读者见面，同时也希望这套书能继续受到读者的关注和青睐。

本书编辑（oyzx_sp@263.net）
2007 年 6 月于清华园

前　　言

在这个数字化时代，全世界有千千万万个网站昼夜不停地提供服务。PHP+MySQL 这一对 Web 应用开发的利器，在从电子商务到网络办公，从邮件服务（例如 eYou）到大型综合网站（例如 Yahoo！）的广泛应用，都显示了其强大的功能。对于从事 Web 应用开发的从业人员或想进入该领域的未来精英而言，掌握 PHP+MySQL 技术是必需的技能。

为帮助 Web 应用从业者快速掌握 PHP+MySQL 开发技术，笔者精心编著了本书。本书遵循由浅入深，循序渐进的学习规律，充分考虑语言学习的特点，紧密结合开发实例，体系完整地介绍了 PHP 网络程序开发技术。

本书特色

1．笔者多年从事 PHP 编程，深知一个初学者在学习 PHP 这门技术中会遇到哪些实际问题。笔者通过本书总结出了初学者在学习 PHP 时经常出现的问题，并对这些问题一一做了解答。这样的做法会使初学者少走不少弯路。

2．本书是一本基础性书籍，由浅入深地对 PHP 作了全面讲解，涉及 PHP 常用的函数及功能，对每个函数及功能都给出了实例，非常便于操作。

3．在本书的后半部分讲解了 PHP 中常用的模块实例，内容包括计数器程序、网上投票系统、留言板程序、简单的 BBS 系统、博客程序及网上商城系统几个部分。实例结合实际功能需要，符合系统设计要求，并且均是网上流行的应用系统。

4．本书内所列举的代码，不论是只有几行的小程序或是有上千行的大型程序都经过了严格的调试运行，都是可以正常运行的。

5．配多媒体视频演示光盘，让读者快速入门。

本书内容

本书共分为 4 篇。其中第 1 篇 PHP 基础，包括第 1~2 章；第 2 篇 PHP 核心技术，包括第 3~15 章；第 3 篇 PHP 常用模块，包括第 16~19 章；第 4 篇 PHP 综合应用实例，包括第 20~21 章。

第 1 章介绍 PHP 基本知识。通过本章，读者可以了解静态网页和动态网页的区别、什么是 PHP 及 PHP 的发展和特点。本章最后通过一个实例，让读者对 PHP 有一个基本的认识。

第 2 章介绍如何构建 PHP 开发环境。内容包括服务器 Apache、PHP 解释器、数据库服务器 MySQL 和管理器及代码编辑器 EditPlus 的安装和配置。

第 3 章介绍 PHP 常量和变量。通过本章，读者可以学习 PHP 中常量和变量的使用。本章还着重介绍数组的使用。

第 4 章介绍 PHP 运算符和表达式。通过本章，读者可以学习 PHP 中各种运算符的使用和表达式的创建。

第 5 章介绍 PHP 的流程控制语句。这些语句包括判断语句 if...else、while、do...while 及 for 循环语句。

第 6 章介绍 PHP 函数的使用。内容包括函数的构成、定义、调用和递归。

第 7 章介绍 PHP 数据的处理。内容包括数据类型的判断、数学函数的应用、字符串的处理和数组的处理等。

第 8 章介绍 PHP 文件和目录操作。内容包括文件的打开、读取、删除和目录的创建、删除。本章最后还介绍文件操作应用实例——在线相册系统的实现方法。

第 9 章介绍主机信息的获取。内容包括浏览器、IP、时间信息的获取和处理。

第 10 章介绍 PHP 的图像处理。内容包括图像函数的使用、图片水印的生成、直方图的绘制、缩略图的生成等。

第 11 章介绍 Session 和 Cookie 的处理。通过本章学习，读者可以掌握 Session 和 Cookie 的定义和使用，并且掌握网页风格变换的实现方法。

第 12 章介绍正则表达式的使用。内容包括 POSIX 和 Perl 正则表达式的构建和应用。本章最后还将介绍 E-mail 校验和 UBB 中正则表达式的应用。

第 13 章介绍 PHP 面向对象编程的基础知识。内容包括类的封装、继承、重载和类相关函数的使用。

第 14 章介绍 MySQL 数据库的使用。内容包括 MySQL 的使用、SQL 语句的使用。本章最后通过一个学生档案系统，介绍 MySQL 使用流程。

第 15 章介绍 PHP 中人机交互的实现。内容包括表单数据的处理和 URL 值传递的处理。

第 16 章介绍计数器的实现。通过本章，读者可以掌握简单计数器、图形计数器、多用户计数器的实现方法。

第 17 章介绍网上投票系统的实现。该系统实现投票的管理、投票统计等功能，并对防止重复投票做专门的讲解。

第 18 章介绍留言板系统的实现。该系统实现留言数据的文件保存、前台留言和后台管理功能。

第 19 章介绍博客系统的实现。该系统实现用户登录和注册、日志添加和修改、留言管理等功能。

第 20 章介绍 BBS 系统的实现。该系统实现 BBS 的常用功能，如用户注册和登录、主题的显示和回复、论坛分类的管理、帖子的管理和用户管理。

第 21 章介绍网上商城系统的实现。该系统功能包括用户注册和登录、商品的分类、搜索和显示以及后台管理等功能。

读者对象

- ☑ 网站开发者和网页设计人员；
- ☑ 高等院校的老师和学生；
- ☑ 相关培训机构的老师和学员；
- ☑ 毕业设计的学生；
- ☑ 以前使用 ASP 或者 JSP，现在转为 PHP 的开发人员。

本书作者

本书由强锋科技统筹，由陈营辉主笔编写。其他参与编写、资料整理和光盘制作的人员有王龙、王拥东、吴善才、徐砚颖、尹健慧、詹涵林、张薇、张小强、张运端、赵玉荣、郑慧、朱博、朱朝坤、邹小红、陈强、陈燕、丁凤霞、丁礼友、范忠诚、黄俊灿、贾伟、李喜彤、林垚、尚文谊、孙亮亮、唐崇敏、陶则熙等。在此对大家的辛勤工作表示感谢！

技术支持

为方便读者学习本书，笔者为本书专门建立了专有论坛：http://www.douban.com/group/32096/，读者可在此讨论书中或其他相关技术问题。

编著者

2007 年 5 月

目　　录

第 1 篇　PHP 基础

第 1 章　初识 PHP......3
1.1　关于静态网页与动态网页......4
1.1.1　传统的静态网页 HTML......4
1.1.2　动态网页与传统网页的区别......4
1.2　关于 PHP......5
1.2.1　什么是 PHP......5
1.2.2　PHP 的发展历史......5
1.2.3　PHP 与其他 CGI 程序相比较......6
1.3　第 1 个程序——HELLO WORLD！......6
1.3.1　在页面中加入 PHP 代码......7
1.3.2　在 PHP 页中加入注释......8
1.3.3　文件的引用......8
1.4　本章小结......9

第 2 章　PHP 的开发环境及安装......11
2.1　Windows 平台下 Apache 的安装......12
2.2　PHP 解释器的安装......14
2.3　MySQL 的安装......15
2.3.1　MySQL 的安装......15
2.3.2　MySQL 的配置......16
2.3.3　修改 php.ini 以支持 MySQL......17
2.4　安装 phpMyAdmin......17
2.5　EditPlus 的安装......18
2.6　本章小结......19

第 2 篇　PHP 核心技术

第 3 章　PHP 中的常量与变量......23
3.1　PHP 中常量的定义与使用......24
3.1.1　定义使用常量......24
3.1.2　PHP 中的预定义常量......25

3.2 PHP 中的变量 26
3.2.1 PHP 的变量类型 26
3.2.2 转换变量类型 27
3.2.3 变量的使用范围 28
3.3 PHP 的预定义变量 28
3.4 PHP 中的数组型变量 29
3.4.1 数组变量的初始化 29
3.4.2 获取数组中的元素 30
3.4.3 给数组动态增加元素 31
3.4.4 创建多维数组 32
3.5 本章小结 34

第 4 章 PHP 中的运算符与表达式 35
4.1 运算符 36
4.1.1 四则运算符 36
4.1.2 逻辑运算符 36
4.1.3 三目运算符 37
4.1.4 运算符的优先级 38
4.2 表达式 40
4.2.1 表达式中变量的可能值 40
4.2.2 赋值表达式的值 40
4.2.3 递增表达式 41
4.2.4 比较表达式 41
4.2.5 组合的运算赋值表达式 42
4.3 运算符与表达式综合运用实例 42
4.4 本章小结 43

第 5 章 PHP 中的流程控制 45
5.1 if…else 判断 46
5.1.1 简单的 if 判断 46
5.1.2 if…else 判断 47
5.1.3 if…else…else 多重判断 47
5.2 switch…case 多重判断 49
5.3 while 循环 50
5.3.1 单纯 while 判断循环 50
5.3.2 使用 break 跳出循环 51
5.3.3 使用 continue 语句 52
5.4 do…while 循环 53

5.5　for 循环 54
5.6　流程控制综合运用实例 55
5.7　本章小结 56

第 6 章　PHP 中的函数 57
6.1　什么是函数 58
6.2　函数的参数 59
6.3　函数的返回值 61
6.4　PHP 内部函数的使用 63
6.5　PHP 加载外部函数 64
6.6　自定义函数 66
6.6.1　如何自定义函数 66
6.6.2　使用自定义函数 67
6.6.3　函数的动态调用 68
6.6.4　函数的递归 69
6.7　本章小结 70

第 7 章　PHP 的数据处理 71
7.1　怎样判断数据类型 72
7.2　PHP 中常用的数学函数 74
7.2.1　数学计算函数 74
7.2.2　数学三角函数 76
7.2.3　很有用的最值函数 78
7.2.4　产生随机数函数 80
7.2.5　进制转换函数 81
7.2.6　其他数学函数 84
7.3　PHP 中常用的字符串函数 85
7.3.1　取得字符串长度 85
7.3.2　输出字符串 86
7.3.3　截取字符串 88
7.3.4　按特定字符切开字符串 89
7.3.5　去除字符串中的特殊符号 91
7.3.6　转换字符串中的特殊符号为 HTML 标记 92
7.3.7　加入去掉斜线 93
7.3.8　两个字符串的比较 93
7.3.9　改变字符串的大小写 95
7.3.10　其他常用字符串函数 96
7.4　PHP 中常用的数组函数 96

7.4.1　新建一个数组 97
7.4.2　计算数组的元素个数 98
7.4.3　对数组排序及逆排序 99
7.4.4　对数组进行自定义排序 101
7.4.5　移动数组指针 103
7.4.6　获取数组当前元素 104
7.4.7　移去数组中重复的值 104
7.4.8　计算数组中所有值出现的次数 105
7.4.9　合并多个数组 107
7.4.10　其他常用数组函数 109
7.5　本章小结 109

第 8 章　PHP 文件应用 111
8.1　判断文件是否存在 112
8.2　访问文件属性 112
8.3　打开文件 114
8.3.1　用只读方式打开文件 114
8.3.2　用写入方式打开文件 114
8.3.3　用读写方式打开文件 115
8.4　读取文件内容 116
8.4.1　读取文件相应字符 116
8.4.2　按行返回文件内容 117
8.4.3　按行返回文件内容去除 HTML 标记 118
8.4.4　将整个文件内容读入到数组变量中 119
8.5　删除文件 120
8.6　创建目录 120
8.7　删除目录 121
8.8　浏览目录下的文件 122
8.9　关于文件上传 123
8.10　文件操作综合实例：在线相册 126
8.10.1　系统功能 126
8.10.2　相册系统首页面 126
8.10.3　相册系统上传前台页面 128
8.10.4　相册系统上传后台页面 130
8.10.5　相册系统浏览图片详细信息页面 131
8.10.6　相册系统图片删除页面 132
8.10.7　测试相册系统 134
8.11　本章小结 137

第 9 章　用 PHP 获取主机信息 139
9.1　phpinfo()的使用 140
9.2　浏览器相关操作 140
9.2.1　获取访问者浏览器信息 141
9.2.2　获取访问者的 IP 地址 142
9.3　日期时间相关函数 142
9.3.1　检查日期的合法性 142
9.3.2　格式化输出当前日期 143
9.3.3　获得时间及日期信息 145
9.4　本章小结 146

第 10 章　PHP 中的图像处理 147
10.1　图像函数 148
10.1.1　访问图像的属性 148
10.1.2　使用 GD 库函数 149
10.2　图像使用实战 154
10.2.1　使用 GD 库函数创建图像 154
10.2.2　创建图形并在上面画图 155
10.2.3　绘制几何图形更多的探索 156
10.2.4　使用 GD 库函数在图片上写字 158
10.2.5　使用 GD 库函数绘制直方统计图 159
10.2.6　用 GD 库函数创建图像的缩略图 161
10.3　本章小结 162

第 11 章　PHP 中的 Session 与 Cookie 163
11.1　Session 的使用 164
11.1.1　如何使用 Session 164
11.1.2　Session 使用实例 166
11.1.3　使用 Session 的注意事项 166
11.2　Cookie 的使用 167
11.2.1　为什么使用 Cookie 167
11.2.2　怎样使用 Cookie 168
11.2.3　设置 Cookie 生命期 169
11.2.4　Cookie 综合应用实例——网页风格转换 170
11.3　本章小结 176

第 12 章　PHP 中正则表达式的使用 177
12.1　关于正则表达式 178

12.1.1 什么是正则表达式 178
12.1.2 如何使用模式匹配 180
12.2 POSIX 扩展的正则表达式函数 183
12.2.1 替换字符串 183
12.2.2 匹配字符串 185
12.2.3 替换字符串（忽略大小写） 187
12.2.4 匹配字符串（忽略大小写） 187
12.2.5 分割字符串到数组 188
12.2.6 分割字符串到数组（忽略大小写） 190
12.2.7 返回包含指定字符的正则表达式 191
12.3 Perl 兼容的正则表达式函数 192
12.3.1 Perl 兼容正则表达式的使用规范 192
12.3.2 返回匹配数组 193
12.3.3 全局表达式匹配 194
12.3.4 正则表达式匹配 196
12.3.5 转义正则表达式字符 197
12.3.6 用回调函数实现正则表达式的搜索与替换 198
12.3.7 执行正则表达式的搜索与替换 199
12.3.8 用正则表达式分割字符串 201
12.4 正则表达式使用实例 202
12.4.1 构造检查 E-mail 的正则表达式 202
12.4.2 对图像 UBB 代码进行替换 203
12.5 本章小结 205

第 13 章 PHP 面向对象编程 207
13.1 面向对象编程（OOP）基础 208
13.1.1 什么是类（class） 208
13.1.2 为什么要用到类 209
13.2 在 PHP 中创建和使用类 209
13.2.1 自建类——类的封装 209
13.2.2 为类添加属性 212
13.2.3 为类添加方法 213
13.2.4 类的继承 215
13.2.5 类的重载 216
13.2.6 类的引用 218
13.2.7 类的构造函数 219
13.3 PHP 中与类、对象相关的函数 220
13.4 类的具体使用实例 230

13.5　本章小结 232

第 14 章　使用 MySQL 数据库 233

14.1　关系型数据库基础 234
14.1.1　什么是关系型数据库 234
14.1.2　关系型数据库的功能 234
14.2　PHP 中与 MySQL 数据库相关的函数 234
14.3　数据库操作 242
14.3.1　连接 MySQL 服务器 242
14.3.2　连接到服务器并显示可用数据库 243
14.3.3　在服务器上创建新的数据库 244
14.3.4　在选定数据库里创建表 245
14.3.5　如何删除已经存在的库和表 247
14.4　对 MySQL 表进行操作 250
14.4.1　执行 INSERT INTO 语句插入记录 250
14.4.2　执行 SELECT 查询 252
14.4.3　使用表单扩展添加记录功能 253
14.4.4　执行 UPDATE 语句更新记录 256
14.4.5　使用表单扩展更改记录功能 258
14.4.6　执行 DELETE 语句删除记录 262
14.4.7　执行 ALTER TABLE 语句改变表的结构 263
14.5　对 MySQL 表的高级查询 265
14.5.1　使用 WHERE 子句 265
14.5.2　使用 LIMIT 子句对结果进行分页显示 267
14.5.3　使用 ORDER BY 对查询结果进行排序 269
14.6　MySQL 数据库使用实例 271
14.6.1　明确设计目的——学生档案管理 271
14.6.2　连接 MySQL 服务器建立学生档案表 272
14.6.3　显示学生数据页的创建 272
14.6.4　添加记录页的创建 275
14.6.5　修改记录页的创建 278
14.6.6　查找记录页的创建 281
14.6.7　学生档案管理系统实际使用 284
14.7　本章小结 286

第 15 章　用 PHP 实现人机交互 287

15.1　用表单实现人机互动 288
15.1.1　表单元素的组成 288

15.1.2　在普通 Web 页中插入表单 290
15.1.3　更改表单的 ACTION 属性到 PHP 程序 293
15.1.4　表单 METHOD 属性 POST 与 GET 区别 293
15.1.5　用 PHP 作后台处理表单提交数据 293
15.2　URL 参数与 PHP 296
15.2.1　在 PHP 的 URL 地址上加入参数 296
15.2.2　用 PHP 处理提交的参数 296
15.3　表单使用实例 298
15.4　本章小结 302

第 3 篇　PHP 常用模块

第 16 章　计数器程序 305
16.1　简单计数器 306
16.1.1　计数器的原理 306
16.1.2　设计算法 306
16.1.3　代码实现 306
16.2　图形化计数器 307
16.2.1　设计算法 307
16.2.2　用图片替代文本 307
16.2.3　代码实现 308
16.3　添加“防止恶意刷新”功能 309
16.3.1　设计算法 309
16.3.2　代码实现 309
16.4　多用户计数器 310
16.4.1　多用户的原理 310
16.4.2　实现方法 311
16.4.3　代码实现 311
16.5　本章小结 312

第 17 章　网上投票程序 313
17.1　投票程序的原理 314
17.2　本实例的特点 314
17.3　投票实例数据表的设计 314
17.4　代码实现 315
17.4.1　准备工作 316
17.4.2　创建显示所有投票项的页面 317
17.4.3　创建添加投票记录页面 318

17.4.4 创建删除投票项的页面 322
17.4.5 创建显示投票项页面 323
17.4.6 创建选择项提交处理页面 326
17.5 测试程序 327
17.6 如何防止重复投票 330
17.7 本章小结 331

第 18 章 文本留言板程序 333
18.1 留言板分类 334
18.1.1 文本型留言板 334
18.1.2 数据库型留言板 334
18.2 文本留言板的实现原理 335
18.3 文本留言板实例的组成 336
18.4 文本留言板代码的实现 336
18.4.1 配置文件的建立 336
18.4.2 显示提交留言文件建立 337
18.4.3 管理入口页的创建 341
18.4.4 修改留言页面的创建 343
18.4.5 删除留言页面的创建 345
18.5 调试运行留言板 347
18.6 进一步完善 349
18.7 本章小结 349

第 19 章 PHP 博客程序 351
19.1 什么是博客 352
19.2 简单博客程序的功能 352
19.3 制作前的准备工作 352
19.3.1 配置文件的创建 353
19.3.2 安装文件的创建 353
19.3.3 头文件的创建 357
19.3.4 侧边文件的创建 358
19.4 日志显示模块 361
19.4.1 主显示页面的创建 361
19.4.2 日志显示页面的创建 363
19.4.3 单条日志详细信息页面的创建 367
19.5 管理模块 370
19.5.1 管理员/用户登录页面 370
19.5.2 更改用户注册信息 373

19.5.3　更改注册用户密码页面 376
19.5.4　修改已经存在的日志类别 379
19.5.5　签写新的日志页面 381
19.5.6　已经存在的日志管理页 384
19.5.7　留言的管理 387
19.5.8　注册用户的管理 389
19.5.9　添加新类别页面 391
19.5.10　留言显示与发表页面 393
19.5.11　新用户注册页面 396
19.5.12　用户退出登录页面 400
19.6　进一步完善 400
19.7　关于 RSS 内容聚合 401
19.8　本章小结 402

第 4 篇　PHP 综合应用实例

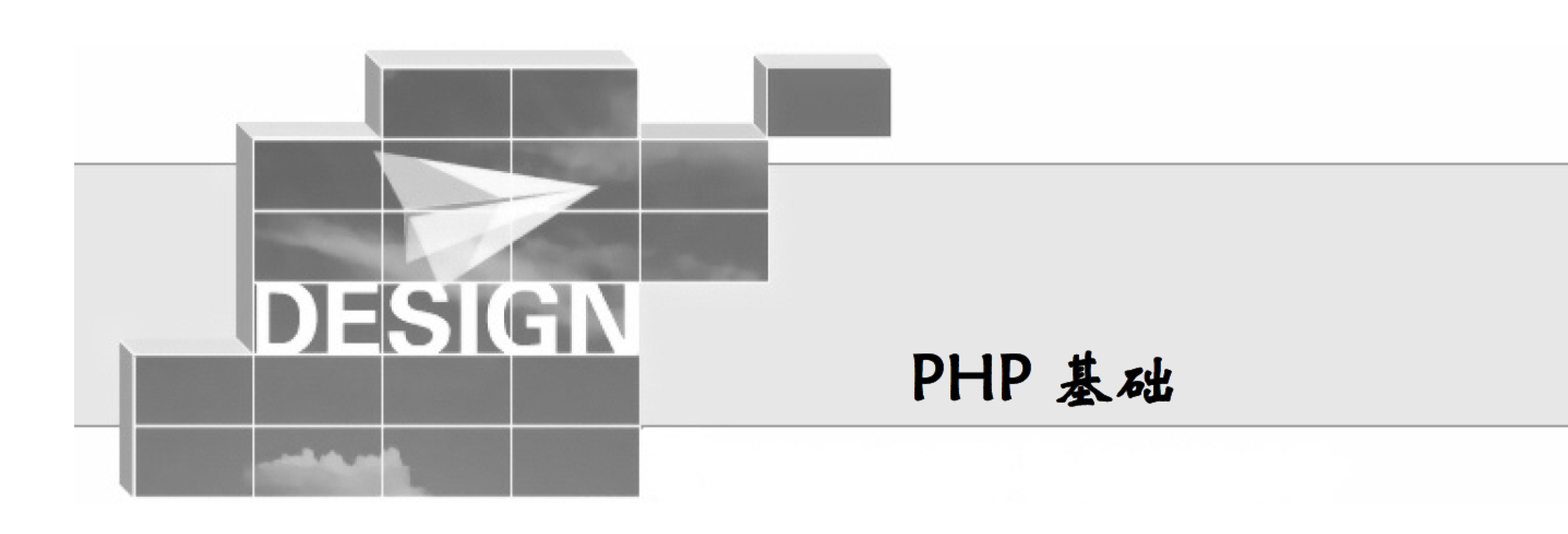
第 20 章　简单的 BBS 系统 405
20.1　设计数据库表 406
20.1.1　用户数据表的设计 406
20.1.2　论坛分类数据表的设计 407
20.1.3　贴子数据表的设计 408
20.2　准备工作 408
20.2.1　配置文件的创建 409
20.2.2　安装文件的创建 409
20.3　用户注册与登录 416
20.3.1　用户注册页 416
20.3.2　注册的后台处理 420
20.3.3　用户登录页面 422
20.3.4　登录出错及处理 425
20.4　论坛首页、主论坛、分论坛显示文件的创建 426
20.4.1　论坛首页显示文件的创建 426
20.4.2　主论坛显示文件的创建 430
20.4.3　分论坛显示文件的创建 432
20.5　主题的显示与回复 435
20.5.1　发表新的主题 435
20.5.2　现有主题的显示 438
20.5.3　主题的回复 441
20.6　论坛分类的管理 442

20.6.1　为论坛增加新的类别 442
20.6.2　更改现有类别 446
20.7　贴子的管理 450
20.7.1　编辑贴子 450
20.7.2　删除贴子 455
20.7.3　防掘墓功能 457
20.8　用户管理 458
20.8.1　用户信息的显示 459
20.8.2　普通信息的修改 461
20.8.3　用户密码的修改 465
20.8.4　用户退出页面 468
20.9　进一步完善 470
20.10　本章小结 471

第 21 章　网上商城全站系统 473
21.1　系统分析 474
21.2　设计数据库表结构 474
21.2.1　用户表的设计 474
21.2.2　图书类型表的设计 475
21.2.3　图书记录表的设计 475
21.2.4　订单记录表的设计 476
21.2.5　销售记录表的设计 476
21.3　准备工作 477
21.3.1　配置文件的创建 477
21.3.2　安装文件的创建 477
21.3.3　头文件的创建 482
21.4　用户的注册与登录 483
21.4.1　用户注册 483
21.4.2　用户登录 486
21.5　前台显示界面 490
21.5.1　首页面的实现 490
21.5.2　图书列表页面的实现 495
21.5.3　按种类查看页面 496
21.5.4　搜索图书页面 499
21.5.5　查看图书详情页面 502
21.6　购物车的实现 505
21.6.1　查看当前购物车 505
21.6.2　查看用户历史订单 508

21.7　管理功能的实现 510
21.7.1　更改用户信息 510
21.7.2　更改用户密码 512
21.7.3　为图书添加新的分类 515
21.7.4　修改已经存在的分类 518
21.7.5　增加新的图书 522
21.7.6　修改已有图书 527
21.7.7　查看、处理所有未处理订单 533
21.7.8　销售记录的查看 535
21.7.9　当前登录用户退出页面 537
21.8　本章小结 537

第 1 篇



第1章　初识PHP

第2章　PHP的开发环境及安装

第 1 章

初识 PHP

- 静态网页与动态网页
- 什么是 PHP 及其发展过程
- 如何加入 PHP 代码
- PHP 的 3 种注释

要想学习 PHP，先要了解到底什么是 PHP，如何使用 PHP。本章将回答这些问题。通过对本章的学习，将使读者对 PHP 及其发展过程有一个大致的了解。同时将学会如何把 PHP 代码加入到普通 Web 页中，如何为 PHP 代码加入注释以及 PHP 的文件引用的特性。

本章介绍如何编写一个 PHP 文件。在学习本章时有几个问题需要注意：第 1，对<%%>的代码使用方法要谨慎使用；第 2，注意不要使用多行注释；第 3，采用 include 和 require 方法引用文件时，被引用的文件必须存在。

1.1　关于静态网页与动态网页

目前网上的网页有静态网页与动态网页两种形式。在讲这两种网页之前，先了解一下网络构成中的客户机（Browser）与服务器（Server）。服务器是安装有服务软件可以向客户机提供诸如网页浏览、数据库查询等服务的设备。而客户机则与之相反，它通过客户端软件如网页浏览器从服务器上获得如网页浏览、软件下载等服务。简单地讲，服务器就是服务提供者，而客户机则是服务获得者。

1.1.1　传统的静态网页 HTML

在 WWW 发展的早期阶段，由于受技术条件的制约，服务器提供给用户的网页基本都是静态的 HTML 页。这种网页通常只包含 HTML 标识，没有脚本代码。这类网页在视觉上也可能出现“动”的效果，如通过 GIF 动画、Flash、JavaScript 特效等内容来丰富网页。但是用户每次浏览，该网页的内容都是一成不变的。

静态网页服务的实现流程如下：客户端通过浏览器向服务器发出请求，服务器根据请求从服务器端的网页中选出合适的页面发送给客户端浏览器。这个过程中所发送的页面都是事先编辑好的，它并不能自动生成。静态网页的实现模式如图 1.1 所示。

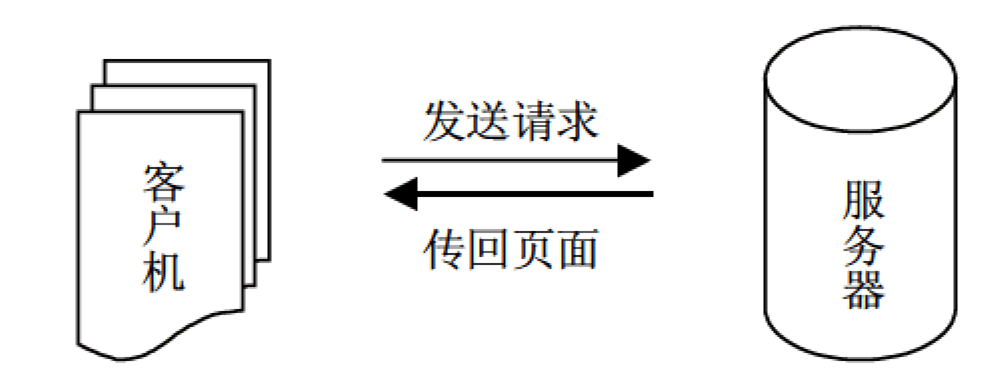


图 1.1　静态网页的实现模式

静态网页有以下几个特点：

- ❑ 静态网页不能自动更新。由于不能自动更新，所以如果要对静态页面进行更新，就要重新编写 HTML 页然后再上传。因此静态网页制作和维护的工作量相当大。
- ❑ 静态网页的内容不随浏览用户、浏览时间等条件的变化而变化。无论何人、何时、何地浏览网页，它的内容都是一成不变的（不包括使用 JavaScript 实现一些简单特效）。
- ❑ 静态网页一经发布，无论浏览者浏览与否，它都是实实在在存在的一个文件，都对应一个 URL（统一资源定位符，指的是 Internet 文件在网上的地址）。
- ❑ 用静态网页实现人机交互有相当大的局限性。由于不能动态生成页面，所以用静态网页来实现人机交互是很困难的，它在功能上有很大限制。

1.1.2　动态网页与传统网页的区别

随着网络技术的不断发展，各种动态网页技术也纷纷显示出它们的不凡魅力。先是早期出现的 CGI，又有现在流行的 ASP、PHP、JSP、ASP.NET、ColdFushion 等。虽然这些动态语言分属于不同公司开发（其中 ASP 与 ASP.NET 同属微软公司），也有着不同的运行环境和使用方法，但它们的目标是一样的，就是实现网页浏览者与网页之间的互动。

与静态网页的实现方法不同，动态网页服务的实现过程如下：客户端向服务器提出申请，服务器

根据用户请求，把动态网页内部的代码先在服务器上进行相应的处理，再把生成的结果发送给客户端。其模式如图 1.2 所示。

通过以上分析，与静态网页相比，动态网页有以下几个特点：

- 动态网页在服务器端运行。客户机上看到的只是它的返回结果，不能看到它的源文件。而静态网页则只能通过服务器把网页文件原封不动地传给客户机，本身不进行任何处理。

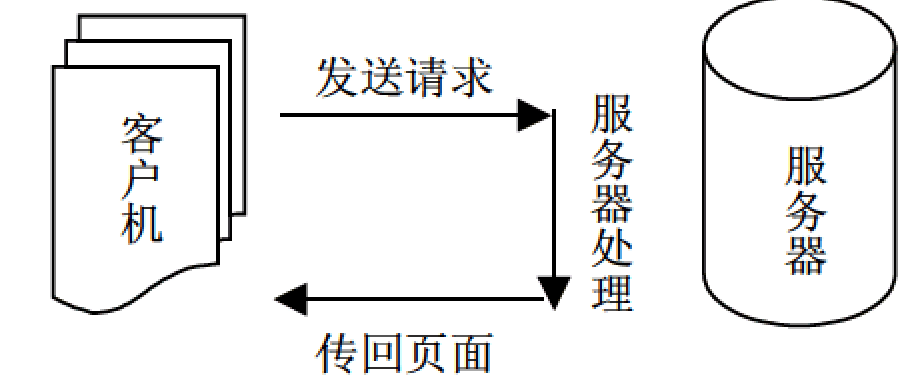


图 1.2　动态网页的实现模式

- 不同的用户、不同时间、不同地点浏览同一个动态网页，根据代码处理结果不同，会返回不同的内容。
- 动态网页只有经用户浏览时才会返回一个完整的网页，而其本身并不是一个独立存在于服务器的网页文件。
- 与静态网页相比，动态网页更容易实现人机交互。与数据库相联系，能实现更为强大的功能。
- 动态网页构建的网站维护起来比静态网页要容易，只需要更新调用的数据（如数据库内容）即可。

1.2　关于 PHP

1.1 节为读者介绍了静态网页与动态网页。而 PHP 就是动态网页技术中的一种。那么到底什么是 PHP，它的发展历史是怎么样的呢？与其他动态网页技术相比，PHP 都有些什么特点呢？本节将为读者来解答这些问题。

1.2.1　什么是 PHP

早期有人将 PHP 解释为 Personal Home Page，即个人主页。也有人将 PHP 称作“PHP：Hypertext Preprocessor”（这是一个递归的简称，简称之中又是简称）。那么到底什么是 PHP 呢？通俗地说，PHP 是一种服务器端、跨平台、HTML 嵌入式的脚本语言。服务器端执行的特性表明它是动态网页的一种。跨平台，则是指 PHP 不仅可以运行在 Linux 系统上，同时也可以运行在 UNIX 或者 Windows 系统上。另外，它还可以很简单地嵌入到普通的 HTML 页中，用户所要做的只是在普通 HTML 页中加入 PHP 代码即可。

1.2.2　PHP 的发展历史

PHP 最初在 1994 年由 Rasmus Lerdorf 进行开发。用户用到的第 1 个版本是在 1995 年发布的 Personal Home Page Tools（PHP Tools）。在这个早期的 PHP 版本中只提供了对访客留言本、访客计数器等简单功能的支持。1995 年中期又发布了 PHP 的第 2 个版本，定名为 PHP/FI（Form Interpreter）。到 1996 年时 PHP/FI 2.0 已经应用于分布在世界各地的 15000 个网站上了。

1997 年开始了第 3 版的开发计划，开发小组加入了 Zeev Suraski 及 Andi Gutmans，而第 3 版就定名为 PHP 3。这个版本的 PHP 具有以下特点：与 Apache 服务器紧密结合；加入了更多的新功能；支持几乎所有主流与非主流的数据库；高速的执行效率等。由于 PHP 3 的这些特性，使得 1999 年中使用 PHP 的网站超过了 150000！

2000 年 5 月，PHP 4 正式发布。它使用了 Zend（Zeev+Andi）引擎，提供更高的性能，还包含了其他一些关键功能，如支持更多的 Web 服务器、HTTP Sessions 支持、输出缓存（output buffering）、更安全的处理用户输入的方法以及一些新的语言结构等。PHP 4 是更有效的、更可靠的动态 Web 页开发工具。在大多数情况下运行比 PHP 3 快，其脚本描述更强大，并且更复杂，最显著的特征是速率比的增加。

2004 年 7 月，PHP 5 问世。无论对于 PHP 语言本身，还是对于 PHP 的用户来讲，PHP 5 的发布都算得上是一个里程碑式的版本。PHP 5 的诞生，使 PHP 编程进入了一个新时代。Zend II 引擎的采用，完备对象模型、改进的语法设计，终于使得 PHP 成为一个设计完备、真正具有面向对象能力的脚本语言。

1.2.3　PHP 与其他 CGI 程序相比较

同样作为服务端编程语言，PHP 与其他 CGI 程序如 ASP、JSP 等相比有其自身的特点。主要表现在以下几个方面：

- ❑ Web 服务器支持方面。PHP 能够被 Apache、IIS 等多种服务器支持，而 ASP 则只能被 Windows 系统下的 IIS、PWS 所支持。
- ❑ 运行平台的支持。PHP 能够很好地运行于 Linux、UNIX、Windows、FreeBSD 等多种操作系统上。而 ASP 只能运行于 Windows 系统上。虽然 JSP 也能在多种系统下得到支持，但必须以有 Java 虚拟机为前提。
- ❑ 脚本语言不同。PHP 本身就是一种编程语言，它是吸收了 C、Java 等语言的综合优势而创建开发的一种新语言。ASP 严格来说并不是一种单纯的编程语言，而是一种网络编程支持环境，它支持 VBScript、JavaScript、Perl 等多种语言，但一般默认使用 VB 作为主要编程语言。而 JSP 使用 Java 编程语言或 JavaScript 作为脚本语言。
- ❑ 数据库支持不同。PHP 通常与 MySQL 数据库结合使用，同时它还支持包括 Oracle、Sybase、ODBC 等多种数据库。ASP 则通常与同属微软公司的 Access、MSSQL 等数据库配合使用。而 JSP 则使用 JDBC 来实现与数据库的连接。
- ❑ 面向对象的支持不同。ASP 基本上是由组件所构成的，而组件是对象的使用模式，因此 ASP 中对象的使用频率非常高，可以说处处都是对象。JSP 是建立在可重用的、跨平台的组件之上的，所以它的面向对象特性也非常明显。在 PHP 5 出现以前，PHP 系列基本上是属于面向过程的，PHP 5 的出现改变了这种状况，真正实现了面向对象。

1.3　第 1 个程序——HELLO WORLD！

学习一门新的编程语言，都需要从最基本的程序开始。根据约定俗成的、最基本的程序就是

"HELLO WORLD!"。这一节，就来介绍怎么在 PHP 编程环境中实现这一个最基本的程序。（读者在学习本节之前先保证已经构建了 PHP 运行环境，如果没有请参看本书第 2 章的相关内容）

1.3.1　在页面中加入 PHP 代码

PHP 是一种可嵌入的语言。也就是说，它可以很方便地加入到一般常见的 HTML 页中。用户请求 PHP 文件时，相关的 PHP 代码先在服务器端解释执行，生成新的 HTML 信息，再连同原有的 HTML 代码一起发送给用户。下面通过一个实例来讲解怎样向普通 HTML 页中加入 PHP 代码。

```
<html>
<head>
<title>HELLO WORLD!</title>
</head>
<body>
<!--以上为普通 HTML 代码，以下为 PHP 代码-->
<?
echo "HELLO WORLD!";                                    //用 echo 打印字符串
?>
<!--以上为 PHP 代码-->
</body>
</html>
```

把以上代码保存为一个 PHP 文件"1-1.php"。.php 为 PHP 文件，本书后面将多次出现。在 PHP 运行环境下（如果读者还没有 PHP 运行环境，请参看本书第 2 章）执行以上代码，结果如图 1.3 所示。

单从执行结果来看，上面这段代码确实比单纯的 HTML 代码没有多执行任何东西。而事实上并非如此。页面中显示的"HELLO WORLD!"是经过服务器的解释才转到客户端的。其执行机理如图 1.4 所示。



图 1.3　HELLO WORLD!执行结果

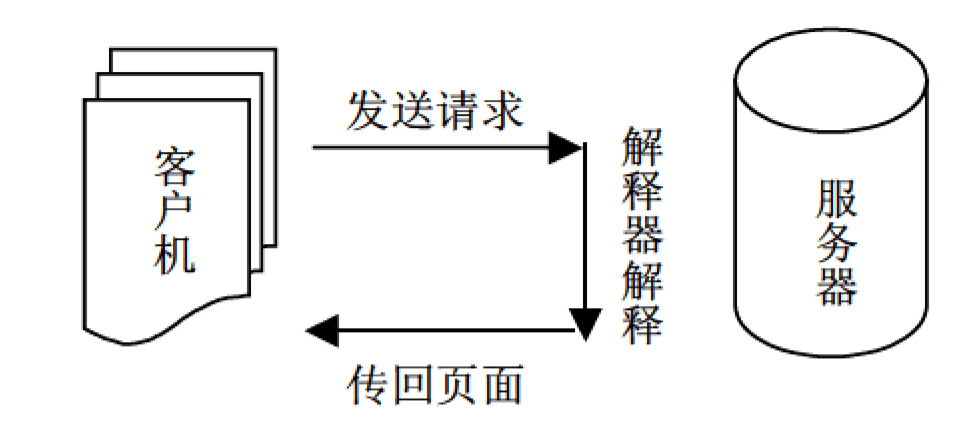


图 1.4　PHP 执行机理

经过上例可以发现，把 PHP 代码加入普通 HTML 页中，只需要在 HTML 页中加入 PHP 代码的标记"<?"、"?>"符号，这些符号的中间即为 PHP 代码。这些代码会先在服务器上被解释，然后再把结果连同普通 HTML 一起返回给客户浏览器。上例中的 PHP 代码即为通过 PHP 中的 ECHO 语句，打印出一段字符串"HELLO WORLD!"。

当然，使用"<??>"标记只是把 PHP 代码加入到普通 HTML 页中的一种方法。除此之外还有以下几种方法：

- ❑　使用<?php ?>标记。
- ❑　使用<script language="PHP"></script>标记。

- ❑ 使用 ASP 语言的<% %>标记，不过这取决于用户服务器的相关配置，有可能不会被解释执行，即原封不动返回客户端。

1.3.2　在 PHP 页中加入注释

每种语言都有自己的注释方法，PHP 也不例外。注释的内容并不被执行，它可以是任何内容。通常，注释的目的是为了向别人说明自己的程序，所以应该是对程序、语句的解释。一个好的程序不仅要有友好的、完善的代码，同时也要有清晰的、易于理解的注释。在程序中加入注释是对自己工作的总结，也是对别人的一种尊重。

下面还采用 1.3.1 节的例子来说明怎样在 PHP 代码中加入注释。

```
<html>
<head>
<title>HELLO WORLD!</title>
</head>
<body>
<!--以上为普通 HTML 代码，以下为 PHP 代码-->
<?
echo "HELLO WORLD!";                                    //使用 ECHO 语句打印字串
/*
PHP 对多行内容的注释
这里有多行内容
*/
?>
<!--以上为 PHP 代码-->
</body>
</html>
```

通过上例知道，为 PHP 代码加入注释，通常有两种方法：一种是单行注释使用“//”标记；另一种是多行注释使用“/* */”标记

注意：在使用多行注释标识的时候，一定不要使用多重注释。像下面注释的使用，必然会引起错误。

```
<?
/*
这里是注释
/*
又有注释出现了，这是错误的
*/
*/
?>
```

在 PHP 中，多行注释并不支持嵌套。

1.3.3　文件的引用

PHP 支持文件的引用。这意味着用户可以把一些全局变量、专用函数放到专门的文件中，需要的

时候，来引用这个文件。文件被引用，就可以使用其中的变量和函数了，就像在一个 PHP 文件中一样方便。PHP 有两种引用文件的方法：require 和 include。下面仍通过一个实例来实际说明这两种方法。

```
<?
$string="HELLO WORLD!";                        //定义变量
?>
```

这段代码只是定义了一个字符串变量。把上面的代码保存为“1-2.php”。

```
<html>
<head>
<title>使用 include 引用文件</title>
</head>
<body>
<?
include("1-2.php");                        //使用 include 引用文件
echo $string;
?>
</body>
</html>
```

把上面的代码保存为“1-2.php”，与 string.php 放在同一个目录下。在 PHP 环境里运行 test.php，执行结果和图 1.3 是一样的。

同理还可以用

```
require "文件名";
```

来引用文件，使用效果与

```
include("文件名");
```

是一样的。

1.4　本 章 小 结

本章介绍了 PHP 的基本知识。第 1 节介绍了静态网页与动态网页；第 2 节主要讲解了什么是 PHP，PHP 的发展历史；第 3 节讲解了怎样在页面中加入 PHP 代码及注释和 PHP 对文件的引用。通过对本章的学习读者对什么是 PHP 及怎么使用 PHP 应该有一个大致的了解了。在实际使用时要注意以下问题：

- 加入 PHP 代码的方法<%%>应谨慎使用，因为如果服务器配置不支持这种方法，则相应的代码不会被执行，而是原封不动地显示给用户。
- 嵌套使用多行注释标记会导致错误。
- 不管用何种方法引用文件，被引用的文件必须是存在的文件。如果文件不存在，也会导致错误。

第 2 章 PHP 的开发环境及安装

- Apache 的安装
- PHP 解释器的安装
- 安装 MySQL 数据库
- 其他辅助工具的安装

通过第 1 章的学习可知，PHP 是一种服务器端编程语言。所以要想运行 PHP 代码，必须有相应的服务器环境及解释器。PHP 能够在多种服务器环境上运行，但是 PHP 的"黄金搭配"还是 PHP+Apache+Linux。但是作为通用操作系统，Linux 远没有 Windows 那么流行。所以本书的环境就采用 PHP+Apache+Windows 这样的形式。本章将介绍怎样在 Windows 操作系统下安装、配置 PHP 的运行环境。

调试 PHP 程序需要安装以下组件。

- ❑ Apache：运行 Web 页面的服务器程序。
- ❑ PHP：PHP 程序的解释器。PHP 页面会先通过该解释器解释再发送给用户。
- ❑ MySQL：MySQL 数据库程序。调试数据库程序的必备服务。
- ❑ phpMyAdmin：用 PHP 编写的管理 MySQL 数据库的程序。使用该程序可以有效管理 MySQL 数据库。
- ❑ EditPlus：PHP 文件的编辑器。可以编辑任何二进制文件。

2.1　Windows 平台下 Apache 的安装

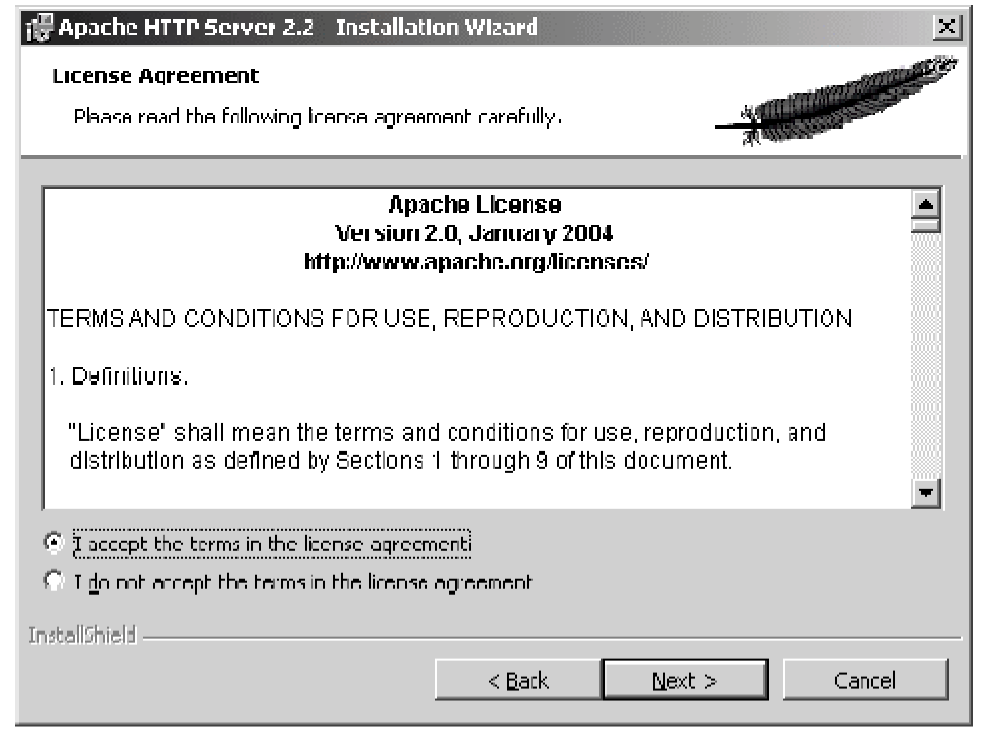
Apache 是运行 PHP 程序最好的服务器系统。通常情况下，Apache 都是运行于 Linux 操作系统上的，但 Apache 也有 Windows 版。本书采用 Apache 的 Windows 版来搭建服务器环境。

Apache 可以在其官方网站 http://www.apache.org 上下载。本书使用 apache_2.2.3 for win32 版本（其实不同版本的区别对初学者来说不是太大）。下载到的是一个 Msi 文件。双击即可安装。最简单的方法是采用一路回车法。其中有几项要向读者作介绍。Apache 的安装共分以下几步：

（1）首先弹出的是欢迎使用界面。单击 Next 按钮，弹出 License Agreement 对话框，如图 2.1 所示。

（2）选中 I accept the terms in the license agreement 选项，单击 Next 按钮，进入下一个对话框。

（3）该对话框对 Apache 简单介绍。单击 Next 按钮，弹出 Server Information 对话框，如图 2.2 所示。



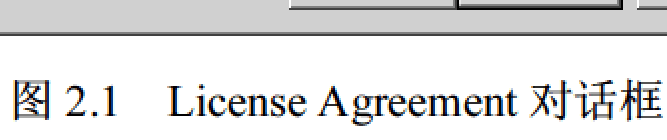


图 2.1　License Agreement 对话框

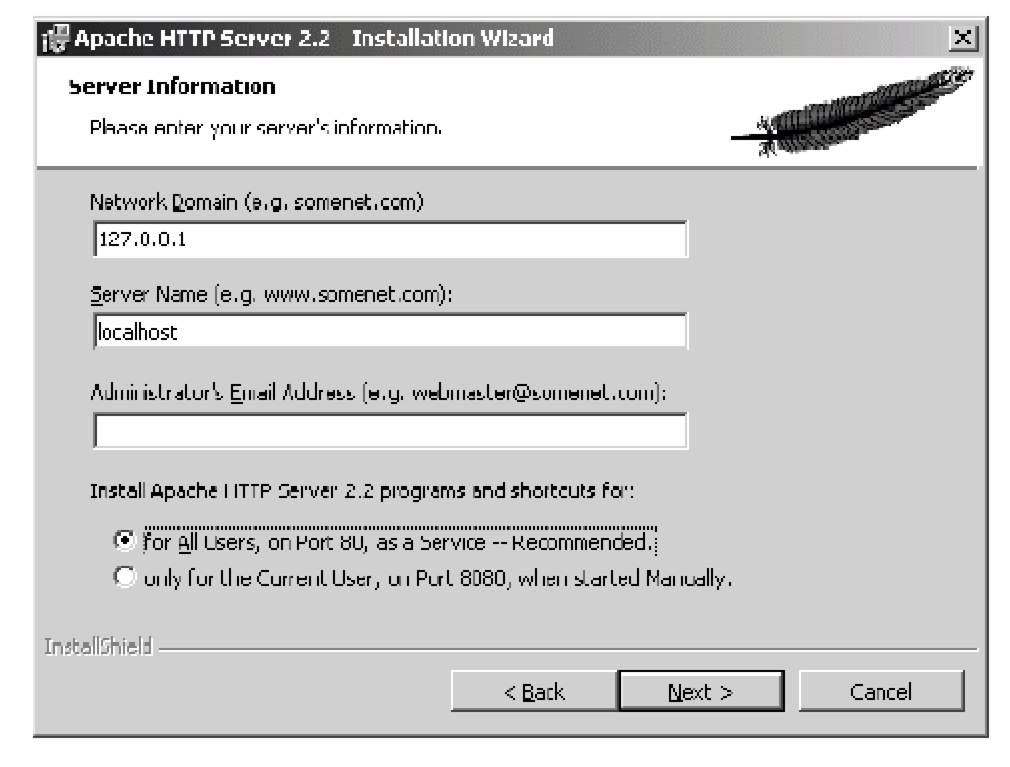


图 2.2　Server Information 对话框

（4）在 Network Domain 文本框中输入域名。如果只是在本机上调试程序，而并不是想架设一台网络服务器，则只需填入本机 IP（127.0.0.1）即可。在 Server Name 文本框中输入服务器的名字，这里输入“localhost”。在 Administrator's Email Address 文本框中输入网络管理员的电子信箱。for All Users, on Port 80, as a Service-Recommended. 选项为允许所有用户使用 Apache 服务；only for the Current User ,on Port 8080, when started Manually. 选项只允许当前用户使用 Apache 服务。这里选中第 1 项。单击 Next 按钮，弹出 Setup Type 对话框，如图 2.3 所示。

（5）Typical 选项为标准安装，Custom 选项为自定义安装，这里选中 Custom 选项。单击 Next 按钮，弹出 Custom Setup 对话框，如图 2.4 所示。

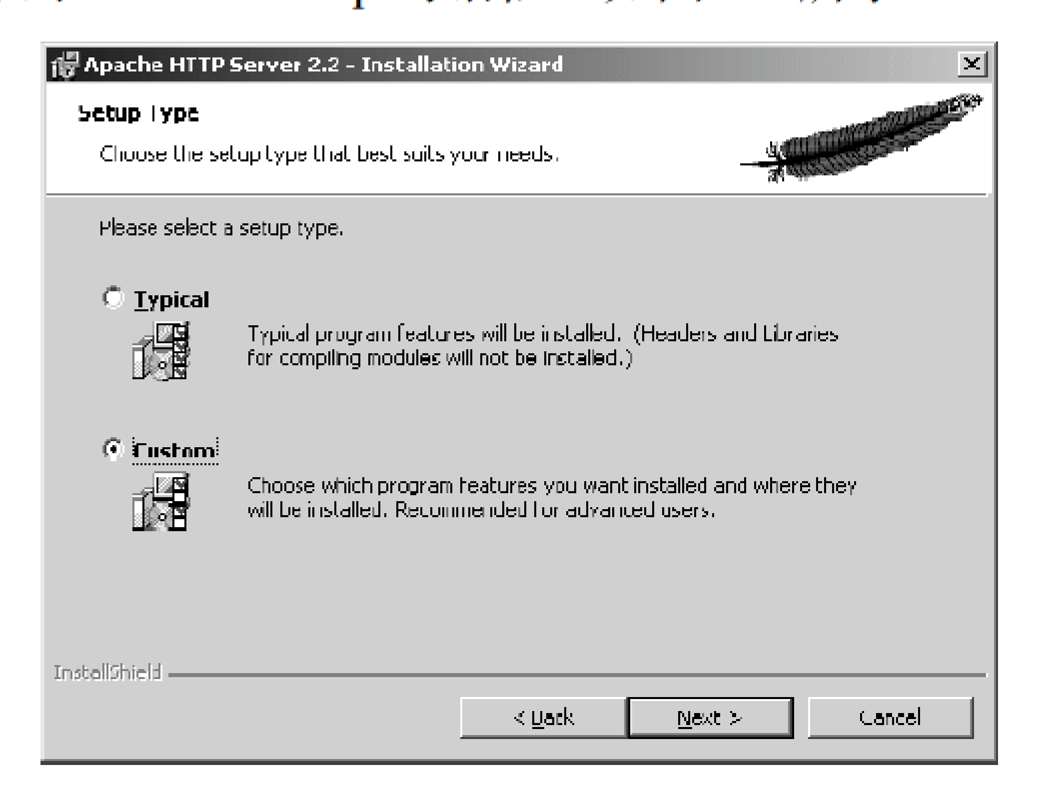


图 2.3　Setup Type 对话框

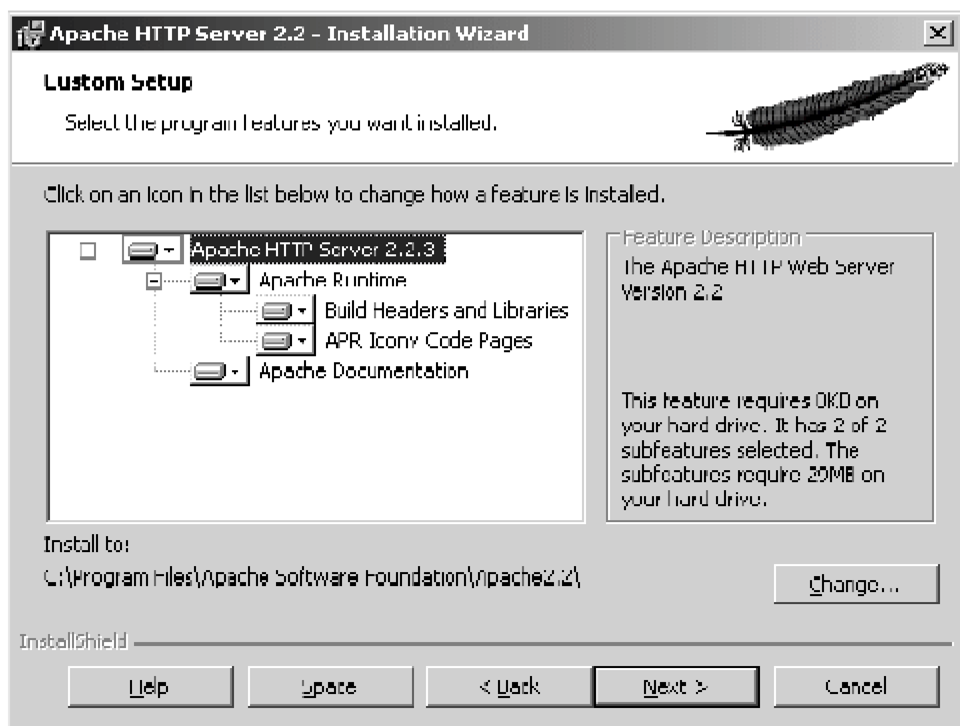


图 2.4　Custom Setup 对话框

（6）单击 Change 按钮，选择安装目录，默认为 C:\Program Files。为了调试方便，这里选择 C:\Apache。单击 Next 按钮，弹出 Ready to Install the Program 对话框，如图 2.5 所示。

（7）单击 Install 按钮开始进行安装。安装结束后，弹出 Installation Wizard Completed 对话框，如图 2.6 所示。

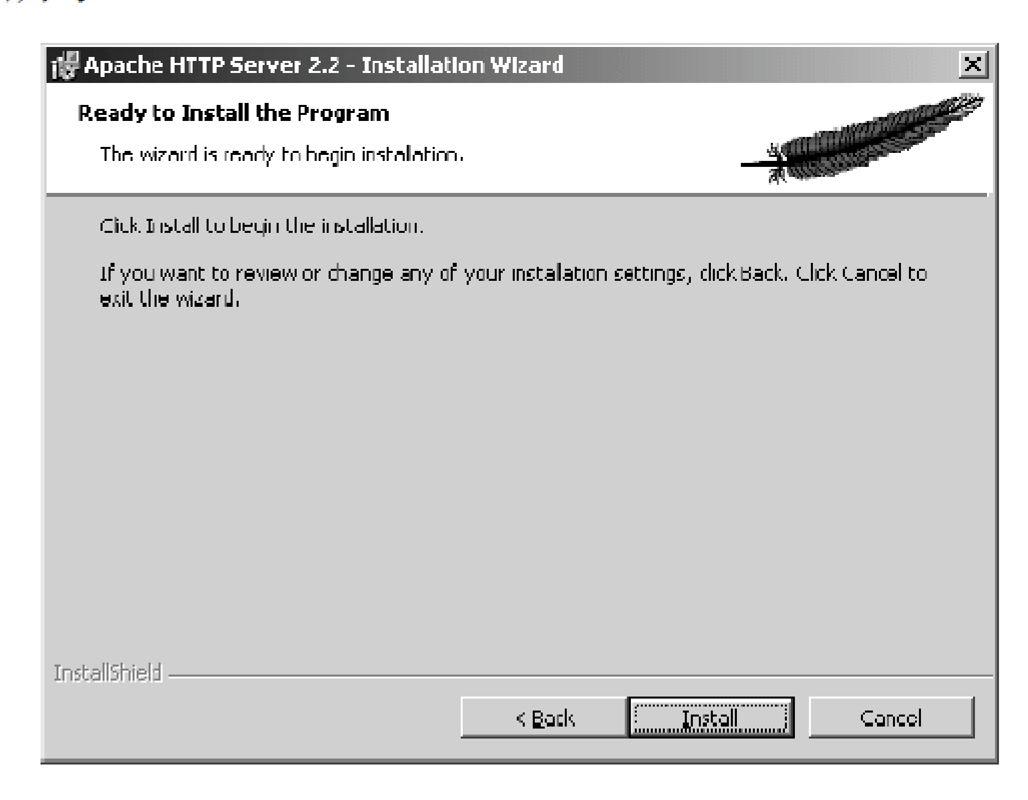


图 2.5　Ready to Install the Program 对话框

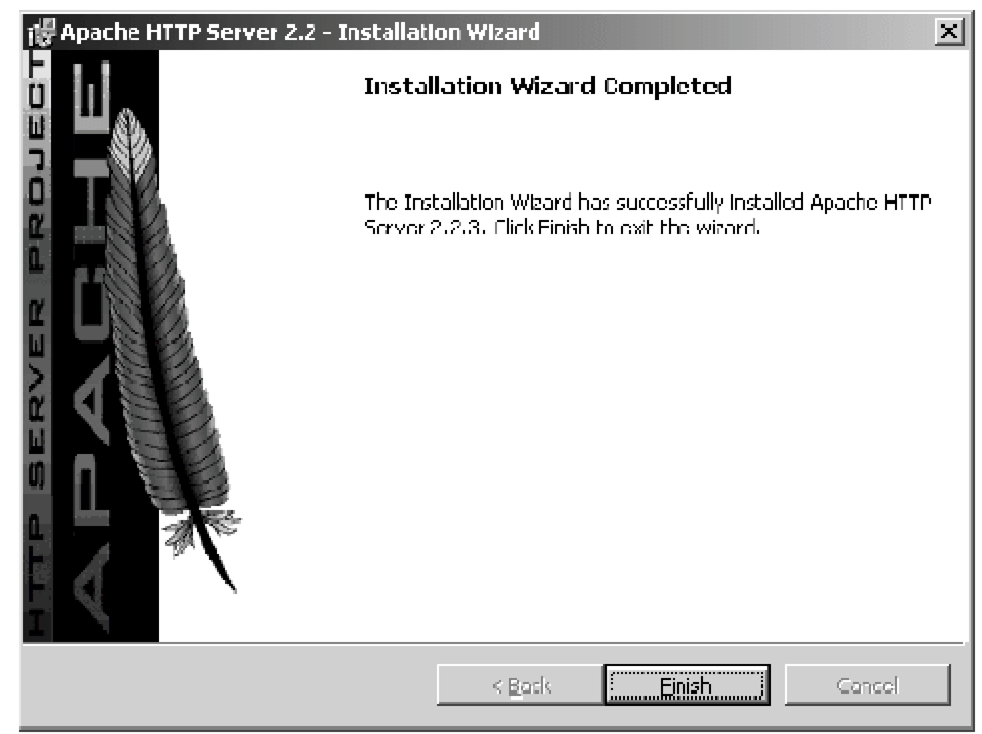


图 2.6　Installation Wizard Completed 对话框

（8）该对话框提示用户 Apache 已经安装成功，单击 Finish 按钮完成安装。

安装成功后，打开 IE 输入“http://127.0.0.1”，结果如图 2.7 所示。至此，Apache 的安装过程全部结束。



图 2.7　测试成功 Apache 正常工作

2.2　PHP 解释器的安装

这一节来完成 PHP 的安装及相关配置。因为 Apache 只支持传统的 HTML，要想体验 PHP 的魅力，必须要安装 PHP 解释器。与 Apache 相比，PHP 的安装就要简单一些，因为 PHP 解释器通常是以 ZIP 包的形式提供，只需在解压缩后进行相关配置即可。

PHP 解释器可以从其官方网站：http://www.php.net/上获取。这里选择 php-5.1.6 的 Win32 版。该版本提供有两种形式，一种是安装包的形式，另一种是 ZIP 包的形式。这里选用 ZIP 包。把下载的 ZIP 压缩包解压到 C 盘 PHP 目录下。解压缩后，需要做以下几步工作。

（1）打开 C:\php 中的文件 install.txt，找到如下内容：

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php

# For PHP 4
Action application/x-httpd-php "/php/php.exe"

# For PHP 5
Action application/x-httpd-php "/php/php-cgi.exe"
```

把这段内容复制到 Apache 的配置文件 httpd.conf（该文件位于 C:\Apache\conf 下）。

（2）右击 Windows 状态栏的图标，选择 open apache monitor 命令，打开 Apache 管理器，如图 2.8 所示。

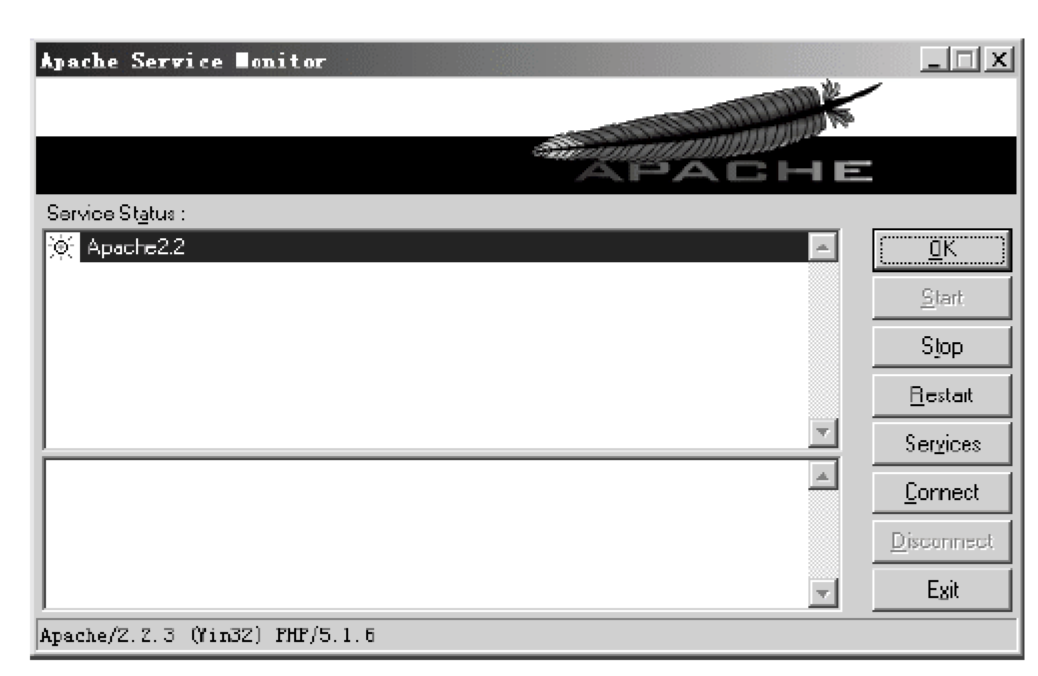


图 2.8　Aapache Service Monitor

（3）单击 Restart 按钮，重启 Apache 服务。

（4）把文件 php.ini-dist 改名为 php.ini。

（5）将 php 目录里的 php5ts.dll，libmysql.dll 复制到系统目录 C:\windows 中。

（6）到 C:\php\ext 下将 php_gd2.dll、php_mysql.dll 这两个文件复制到系统目录 C:\windows\system32 里。

（7）重启 Apache 服务器。

下面测试 Apache 对 PHP 的支持。在 C:\Apache\htdocs 目录下，新建文件 2-1.php，加入如下代码：

```
<?
phpinfo();
?>
```

在浏览器中输入“http://localhost/2-1.php”，结果如图 2.9 所示。出现该界面说明 PHP 安装成功。

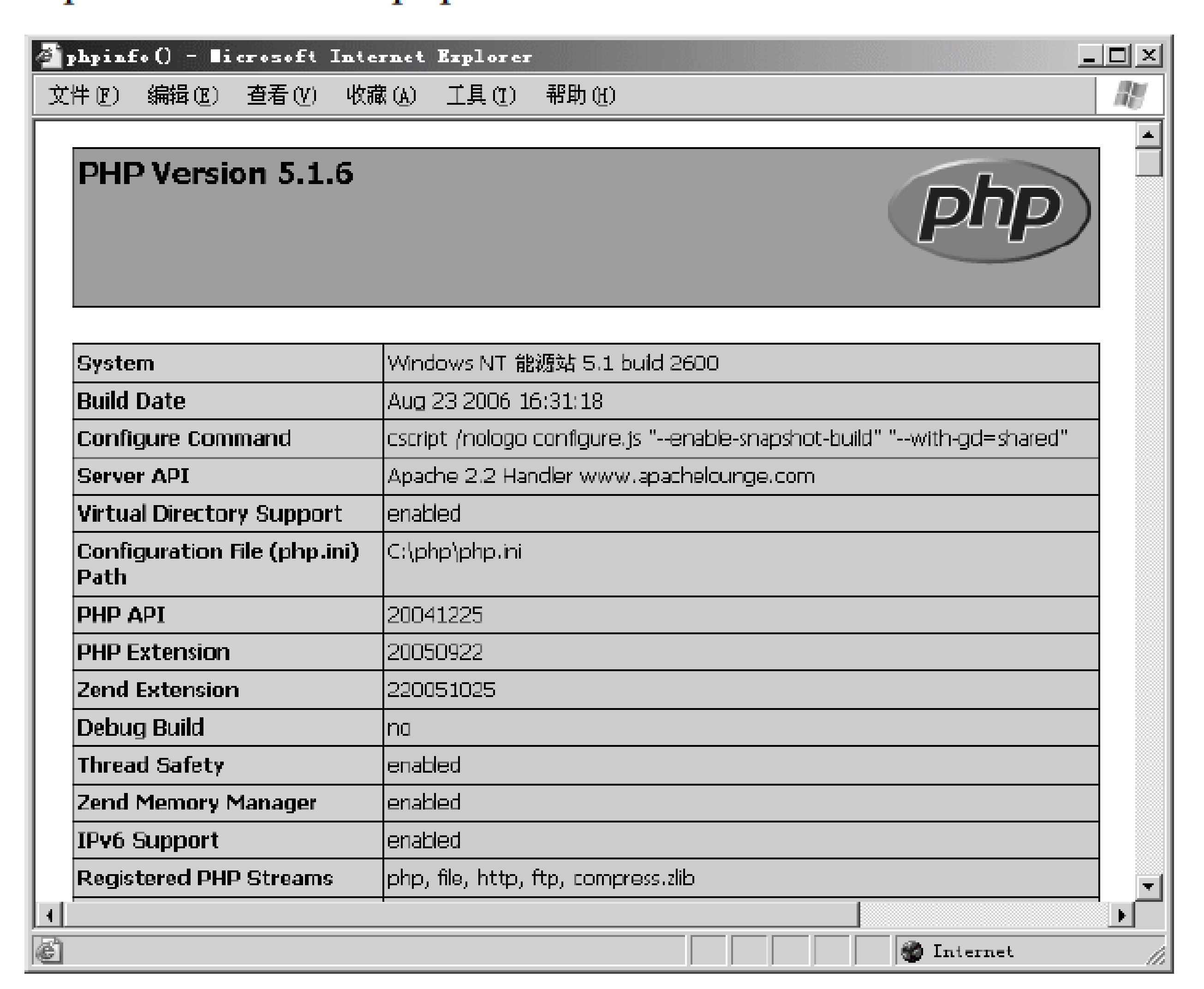


图 2.9　test.php 执行结果

2.3　MySQL 的安装

PHP 通常与 MySQL 数据库配合使用，所以学习 PHP 就得必须学习 MySQL 数据库技术。因此，读者需要安装相应的 MySQL 数据库。这一节来介绍如何获取与安装 MySQL 数据库。

2.3.1　MySQL 的安装

读者首先应到 MySQL 的官方网站 http://www.mysql.com 下载 MySQL 的 Windows 版本。通常下载的是一个 Msi 的安装包，直接双击运行该安装程序即可。整个安装过程共分以下几步：

（1）首先弹出欢迎界面，单击 Next 按钮，进入 Setup Type 对话框，如图 2.10 所示。

（2）MySQL 提供 3 种安装模式：Typical 选项为标准安装；Complete 选项为完全安装；Custom 选项为自定义安装。这里选择 Custom 选项，单击 Next 按钮，弹出 Custom Setup 对话框，如图 2.11 所示。

（3）单击 Change 按钮，更改安装目录为 C:\MySQL。单击 Next 按钮，弹出安装确认界面。

（4）确认安装后，单击 Install 按钮，开始安装。

（5）在安装过程中会弹出 MySQL.com Sign-Up 对话框，如图 2.12 所示。

（6）选中 Skip Sign-Up 选项，单击 Next 按钮即可。

（7）安装完成，弹出 Wizard Completed 对话框，如图 2.13 所示。

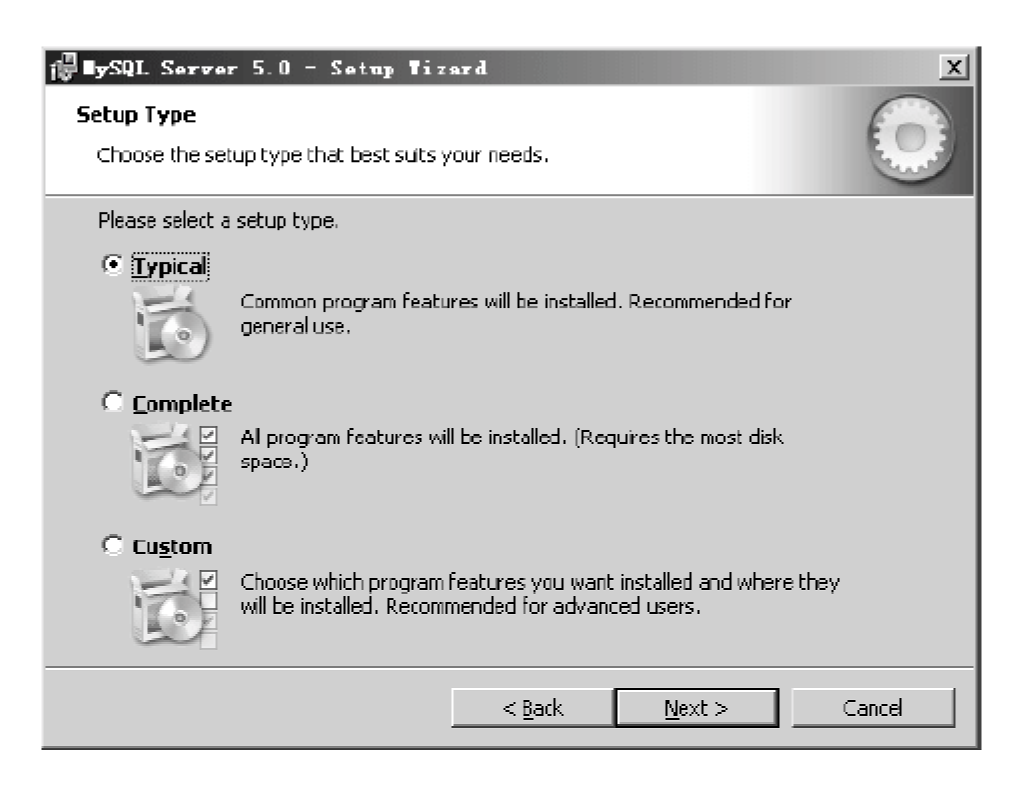


图 2.10　MySQL 安装第 2 步

图 2.11　Custom Setup 对话框

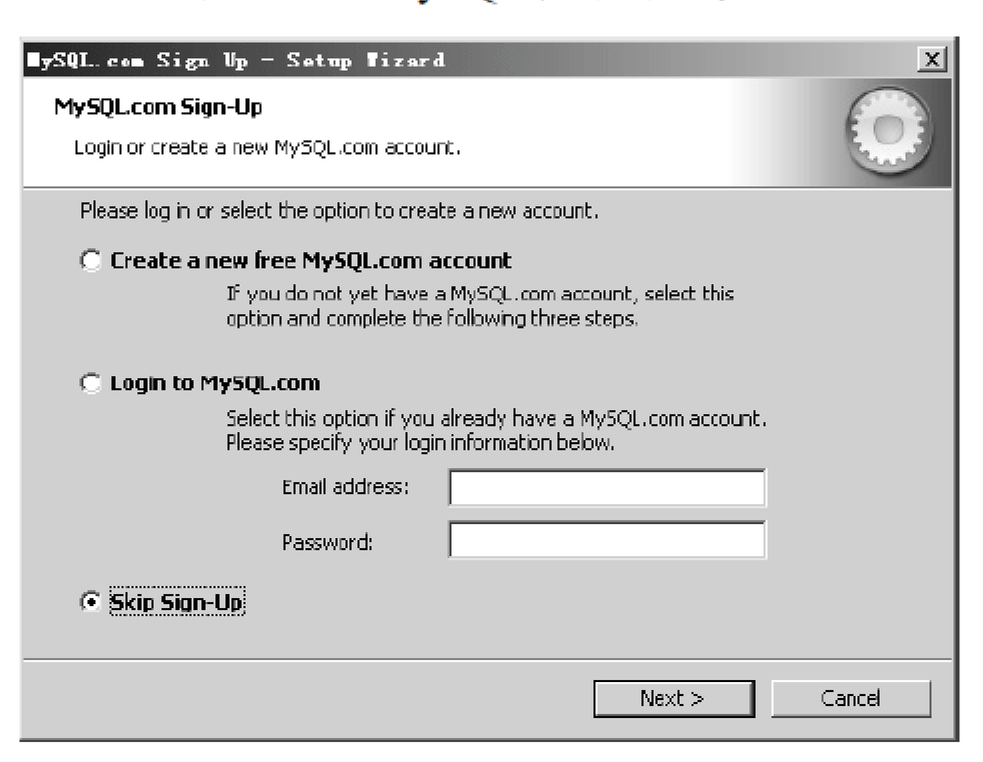


图 2.12　MySQL.com Sign-Up 对话框

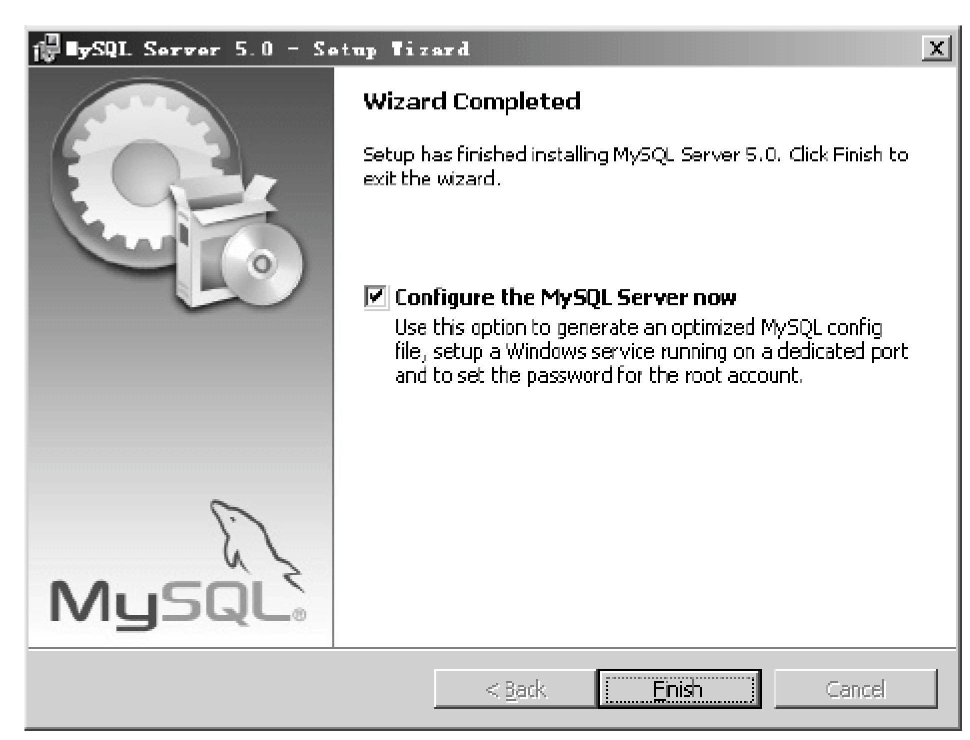


图 2.13　MySQL 安装第 5 步

选中 Configure the MySql server now 选项，单击 Finish 按钮，完成安装过程，开始对 MySQL 服务进行配置。

2.3.2　MySQL 的配置

在安装完成后，开始对 MySQL 服务进行配置。整个配置过程有以下几步：

（1）首先弹出欢迎界面，单击 Next 按钮，进入 MySQL Server Instance Configuration 对话框，如图 2.14 所示。

（2）Detailed Configuration 选项为细节配置；Standard Configuration 为标准配置。这里选中 Standard Configuration 选项，然后单击 Next 按钮，进入 MySQL Server Instance Configuration 对话框，如图 2.15 所示。

（3）Install As Windows Service 选项为安装为 Windows 的系统服务；Include Bin Directory in Windows PATH 选项用命令行启动 MySQL。这里只选择第 2 项，然后单击 Next 按钮，进入配置确认对话框。

（4）如果确认无误，单击 Execute 按钮，执行配置。

（5）单击 Finish 按钮，即可完成配置。

至此整个配置过程完成。

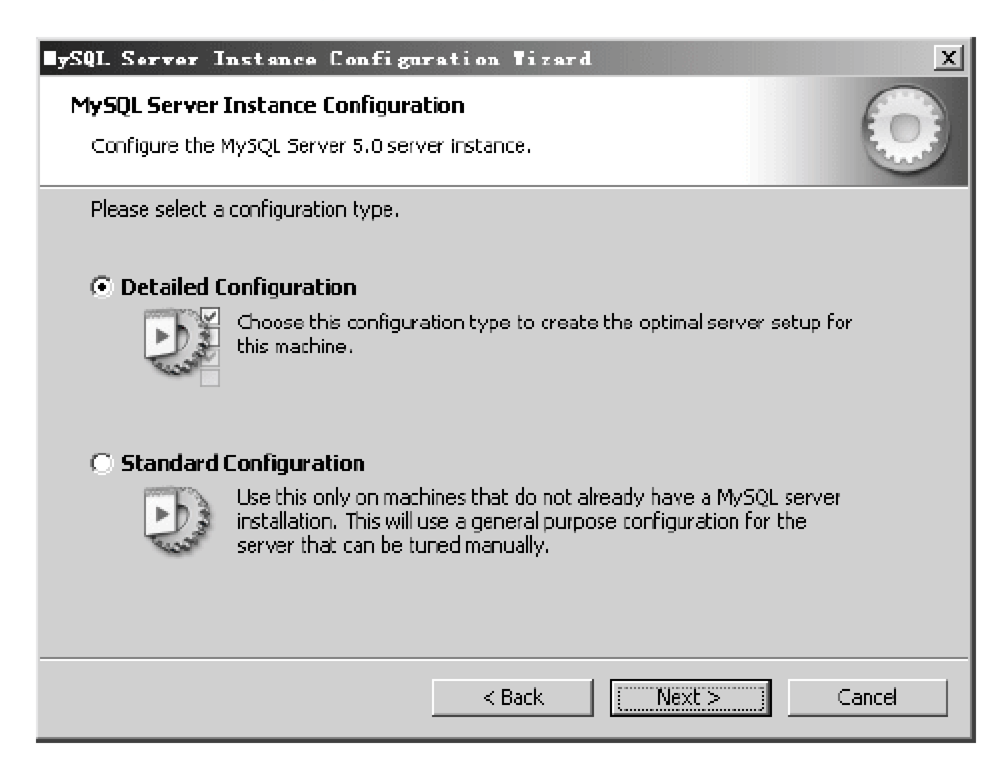


图 2.14　MySQL 配置过程第 2 步



图 2.15　MySQL 配置第 3 步

2.3.3　修改 php.ini 以支持 MySQL

打开 C:\php 目录下的 php.ini 文件。找到

```
;extension=php_mysql.dll
```

这一行内容。去掉前面的分号，保存 php.ini 文件。重新启动服务即可让 PHP 支持 MySQL。在 C:\Apache\htdocs 目录下新建一个 PHP 文件 2-2.php，输入以下内容：

```
<?php
$link=mysql_connect('localhost','root','');          //该函数是进行 MyYQL 主机连接的函数，其中的 root 和空
密码是 MySQL 的用户和密码，请根据自己的情况改好
if(!$link) echo "失败";                              //如果连接失败，输出失败信息
else echo "成功";                                    //如果连接成功，输出成功信息
mysql_close();
?>
```

先开启 MySQL 服务。开启方法：执行 C:\mysql\bin\目录下的 mysqld-nt.exe 文件，MySQL 服务即在后台运行。

然后在 PHP 运行环境下执行 2-2.php，执行结果如图 2.16 所示。

出现如图 2.16 所示结果，说明 MySQL 已经正常运行。



图 2.16　测试 MySQL 是否成功安装

2.4　安装 phpMyAdmin

安装完 MySQL 数据库后，要建库、表，修改各种数据库、表等工作。如果这些工作都在命令行模式下进行，一方面非常麻烦，另一方面也需要有专业的 SQL 数据知识才行。可喜的是，现在已经有

了可视化的 MySQL 数据管理工具 phpMyAdmin。该工具采用 PHP 编写，可以完美运行在各种版本的 PHP 及 MySQL 下。这一节讲解如何安装 phpMyAdmin。

用户可以通过网站 http://www.phpmyadmin.net/home_page/index.php 下载到 phpMyAdmin 的最新版。安装过程分以下几步：

（1）把下载到的压缩包解压到 C:\apache\htdocs\目录下。把目录改名为 phpmyadmin。

（2）复制 libraries/config.default.php 到 phpmyadmin 目录下，并改名为 config.inc.php。

（3）找到 config.inc.php 中的如下代码：

```
$cfg['Servers'][$i]['user']          = 'root';      // MySQL user
$cfg['Servers'][$i]['password']      = '';          // MySQL password (only needed)
```

把其中的内容替换为 MySQL 的用户名和相应的密码即可。

读者还需要测试 phpMyAdmin 是否正常工作。在浏览器中输入 http://127.0.0.1/phpmyadmin/index.php。其执行结果如图 2.17 所示。



图 2.17　phpMyAdmin 执行结果

出现如图 2-17 所示结果说明 phpMyAdmin 正常运行，phpMyAdmin 安装完成。

2.5　EditPlus 的安装

PHP 文件属于二进制文件，所以可以采用所有通用的文本编辑器来进行编辑，如常见的记事本、写字板、Word 等。不过这些程序都不是专为编写程序代码而设计，在编写代码时功能有限。这里为读者推荐一款功能强大的文本编辑软件——EditPlus，用它来编辑 PHP 代码。

用户可以在其官方网站 http://www.editplus.com/上下载到 EditPlus。由于是英文软件，所以建议安装相应的汉化补丁。网上也有很多该软件的汉化补丁。该软件及补丁与普通软件一样，安装过程相当简单，不再赘述。安装后，运行该软件，打开工具菜单下的“参数”对话框，选择【文件】|【设置与语法】命令，弹出如图 2.18 所示对话框。

在右上方的“文件类型”列表框中选择 PHP 选项，然后选中右下方的“关联当前文件类型”复选框。然后单击的“应用”按钮。这样所有的 PHP 文件默认被 EditPlus 打开，完成配置。PHP 的开发环

境全部安装完毕。



图 2.18　EditPlus 的配置

2.6　本 章 小 结

开发环境是进行 PHP 编程的基础，有了相应的环境才能进行开发。本章内容中 Apache 的安装相对简单，而 PHP 安装过程中由于要修改配置文件，所以相对麻烦。而且由于版本兼容性问题，有时需要安装相应补丁才能正常运行，这一点需要引起读者注意。

第 2 篇



PHP 核心技术

第3章　PHP中的常量与变量

第4章　PHP中的运算符与表达式

第5章　PHP中的流程控制

第6章　PHP中的函数

第7章　PHP的数据处理

第8章　PHP文件应用

第9章　用PHP获取主机信息

第10章　PHP中的图像处理

第11章　PHP中的Session与Cookie

第12章　PHP中正则表达式的使用

第13章　PHP面向对象编程

第14章　使用MySQL数据库

第15章　用PHP实现人机交互

第 3 章

PHP 中的常量与变量

- PHP 中的常量
- PHP 中的变量
- PHP 中预定义变量的使用
- PHP 中数组型变量的使用

常量与变量是构成程序的基石，所以每种编程语言中都会有本类语言所对应的常量与变量。作为一门网络编程语言，PHP 也不例外。本章将详细介绍有关 PHP 中的常量与变量的知识。通过本章的学习，读者将会学习到什么是常量、什么是变量；在 PHP 中如何使用预定义常量与变量；如何自定义常量与变量等。

3.1　PHP 中常量的定义与使用

常量是在程序运行中值始终不会发生改变的一类量。在进行 PHP 编程时经常要用到这类数据，如打开文件的文件名、文件的路径等这些系统常量以及用户自定义一些的常量。本小节将介绍 PHP 中的常量。

3.1.1　定义使用常量

常量在使用前必须定义，否则程序在执行时就会出错。在 PHP 中使用 define()函数来定义常量。其语法格式如下：

```
define("Name","value");
```

其中的 Name 为定义常量的常量名，value 为常量代表的值。

下面就通过一个实例来介绍一下 PHP 中的常量的定义与使用。

```
<html>
<head>
<title>PHP 中常量的定义与使用</title>
</head>
<body>
<?
define("STANDARD_H","HELLO WORLD!");            //定义常量 STANDARD_H，并赋值为 HELLO WORLD
echo STANDARD_H;                                //使用 echo 打印常量
?>
</body>
</html>
```

保存以上代码为 PHP 文件“3-1.php”。然后在 PHP 运行环境下运行以上代码，运行效果如图 3.1 所示。



图 3.1　定义使用常量输出结果

常量的命名不是随意的，必须符合一定的规则。PHP 中常量的命名有以下规则：合法的常量名以字母或下划线开始，后面跟着任何字母，数字或下划线。

常量与变量的不同之处体现在以下几个方面：

- ❑ 常量前面没有美元符号$，而变量则必须以美元符号开头。
- ❑ 常量只能用 define()函数定义，而不能通过赋值语句定义。
- ❑ 常量可以不用理会变量范围的规则，可以在任何地方定义和访问。
- ❑ 常量一旦定义就不能被重新定义或者取消定义，并且其值不能发生改变，而变量的值可以随时发生改变。这也是常量与变量最根本的不同。
- ❑ 常量的值只能是标量，即整型、浮点型、字符串 3 种类型。

3.1.2　PHP 中的预定义常量

除了使用自定义常量之外，PHP 还为用户预定义了系统常量，常见的系统常量及其含义如表 3.1 所示。

表 3.1　PHP 中的预定义常量

常　量　名	说　　明
__FILE__	PHP 文件的文件名
__LINE__	PHP 文件的行数
PHP_VERSION	PHP 程序的版本，如 4.0.2
PHP_OS	执行 PHP 解释器的操作系统名称，如 Windows
TRUE	真
FALSE	假
E_ERROR	最近的错误处
E_WARNING	最近的警告处
E_PARSE	剖析语法有潜在问题处
E_NOTICE	发生不寻常但不一定是错误处

以上这些以“E_”开头形式的常数，可以参考 Error_Reporting()函数。下面通过一个实例来实际应用一下 PHP 中的预定义常量。

```
<hmtl>
<head>
<title>PHP 中预定义常量的使用</title>
</head>
<body>
<?
echo "所使用的文件名是：";
echo __FILE__;                                   //输出当前文件名
echo "<br>";                                     //输出 HTML 换行符
echo "文件的行数为：";
echo __LINE__;                                   //输出文件行数
echo "<br>";
echo "PHP 的版本是：";
echo PHP_VERSION;                                //输出 PHP 版本
echo "<br>";
echo "所使用的操作系统为：";
echo PHP_OS;                                     //输出操作系统类型
?>
</body>
</html>
```

保存以上代码为“3-2.php”。在 PHP 运行环境下执行以上代码。执行结果如图 3.2 所示（当然，

实际输出会因操作系统、PHP 版本的不同而有所出入，但大体上是一样的）。

注意：不论是使用自定义常量还是系统预定义常量，大小写都必须一致。如使用系统预定义常量时把大写改为小写，就不能正确返回预定义常量 PHP_VERSION 所定义的 PHP 版本号，而是返回“php_version”这个字符串。

```
<?
echo PHP_VERSION;                              //输出 PHP 版本
echo "<p>";                                    //输出 HTML 换行符
echo php_version;                              //输出字符串
?>
```

把以上这段代码保存为“3-3.php”，在 PHP 执行环境下运行，其执行结果如图 3.3 所示。

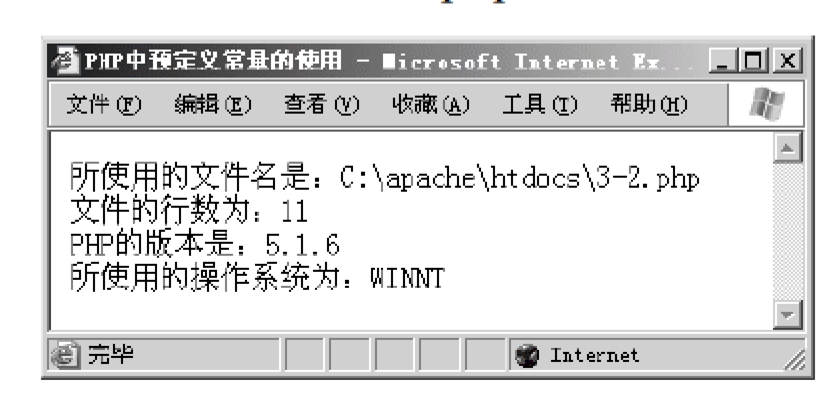


图 3.2　使用 PHP 的预定义常量



图 3.3　预定义常量中的大小写

从图 3.3 所示可以发现，大小写不同，所输出的不同结果。所以，使用系统预定义常量时，一定要注意大小写问题。

3.2　PHP 中的变量

变量指在程序运行过程中值可以随时发生改变的一类值。PHP 是一个弱类型的语言（弱类型语言指在使用变量时不用指定变量的类型，在使用时也没有类型检查的一类编程语言），所以在使用变量时，不用事先指定变量类型，在使用时根据上下文由系统解释器来判断变量的类型。另外 PHP 也不像其他编程要先定义才能使用，变量不用事先定义即可使用。

3.2.1　PHP 的变量类型

PHP 的变量类型有以下几种：整型变量（Integer）、浮点型变量（Double）、字符型变量（String）、数组型变量（Array）和对象型变量（Object）。

- 整型变量在 32 位操作系统中它的有效范围是：−2 147 483 648~+2 147 483 647。要使用 16 进制整数可以在前面加 0x。
- 浮点型变量在 32 位操作系统中的有效范围为：1.7E−308~1.7E+308。
- 字符型变量不同于其他编程语言有字符与字符串之分，在 PHP 中，统一使用字符型变量来定义字符或者字符串。
- 数组型变量是一种比较特殊的变量类型，将在 3.4 节中详细说明。

- 对象变量也是一种比较特殊的变量。在 PHP 5 之前，PHP 面向对象编程的功能还不是很强大。PHP 5 改变了这种状况。类概念的引入，使 PHP 真正成为一种面向对象的编程语言。

要定义一个变量方法很简单就是在该变量名前加上美元符号$。下面的例子就分别定义了两个整型变量和两个字符型变量。

```
<?
$i=0;                          //定义一个整型变量，赋值为 0
$int2=1253;                    //定义一个整型变量，赋值为 1253
$string="a";                   //定义一个字符型变量，赋值为 a
$string1="I'm a teacher!";     //定义一个字符串变量，赋值为“I'm a teacher!”
?>
```

通过以上例子能够发现，在 PHP 中定义一个变量是一件很简单的事情。

3.2.2　转换变量类型

在实际使用 PHP 过程中，有时需要对变量的类型进行强制转换，如要把字符型变量改变为数值型变量，把数值型变为字符型等。在 PHP 中可通过 Settype()函数来设置一个变量的类型。它的使用方式如下所示：

```
settype(mixed var,string type)
```

作用是将变量 Var 的类型设置成 Type。Type 的可能值（即能够转变的类型）为：Boolean（或为 bool，从 PHP 4.2.0 起）、Integer（或为 Int，从 PHP 4.2.0 起）、Float（只在 PHP 4.2.0 之后可以使用，对于旧版本中使用的 Double 现已停用）、String、Array、Object、Null（从 PHP 4.2.0 起）。

如果类型转换成功则返回 True，失败则返回 False。

下面通过一个例子来说明 settype()函数是怎么实现变量类型设置的。

```
<html>
<head>
<title>settype()函数使用例子</title>
</head>
<body>
<?
$foo="5bar";                   //定义一个字符串变量
$bar=true;                     //定义一个逻辑值变量
echo $foo;                     //输出变量$foo
echo "<p>";                    //输出 HTML 回车换行
echo $bar;                     //输出变量$bar
echo "<p>";
settype($foo,"integer");       //重新设置$foo 的类型为整型
settype($bar,"string");        //重新设置$bar 的类型为字符型
echo $foo;                     //重新输出$foo
echo "<p>";
echo $bar;                     //重新输出$bar
?>
```

```
</body>
</html>
```

保存以上代码为“3-4.php”，在 PHP 执行环境中运行以上代码，其执行结果如图 3.4 所示。



图 3.4　settype()函数使用例子执行结果

在使用 Settype()函数前，$Foo 变量值为字符串、$Bar 变量值为逻辑真值。所以打印出它们的值就为：“5bar”、“1”；在使用 Settype()函数后，$Foo 变量值改变为整型数、$Bar 变量值改变为字符串。所以打印的结果就成了“5”、“1”。

3.2.3　变量的使用范围

和其他编程语言一样，PHP 中的变量也有全局变量与局部变量之分。所谓全局变量指在程序运行期间都能使用的变量，而局部变量只在子函数或过程中有效。在 PHP 程序执行时，系统会在内存中保留一块全局变量的区域。实际运用时，可以通过$GLOBALS["变量名称"]的数组调用方法将需要的全局变量调出。不过需要注意的是 PHP 的变量有大小写之分。如果搞错了大小写是不能调出来的。

$GLOBALS 数组是 PHP 程序中比较特殊的变量，不必声明系统会自动匹配相关的变量在里面。在函数中，也不必管$GLOBALS 数组是否已经被声明，就可以直接使用。

和$GLOBALS 变量类似的，还有$php_errormsg 字符串变量。若 PHP 的配置文件 php.ini 中的 track_errors 选项值为 True 的话，使用全局变量$php_errormsg 可以看到错误的讯息。

在 PHP 中，全局变量的有效范围只限于主程序中，不会影响到函数中同名的变量，也就是全局变量与局部变量互不干扰。若要全局变量也能在子函数中使用，就要用到$GLOBALS 数组或是使用 globals 宣告。

例如，在自行开发的函数中，要取得目前执行 PHP 文件的文件名，就可以用$GLOBALS["PHP_SELF"]取出$PHP_SELF 的值。

3.3　PHP 的预定义变量

PHP 在系统中内置了大量与系统、正在运行的 PHP 文件、HTTP 等相关的变量，如表 3.2 所示。了解和使用这些内置变量对提高编程效率有很大帮助。本节将介绍一些常用的 PHP 预定义变量，更多的变量请参考 Phpinfo()函数所列出的内容。

表 3.2　PHP 中的预定义变量

名　　称	作　　用
$_SERVER[PHP_SELF]	当前正在执行的文件名。返回值与document root相关
$_SERVER[REQUEST_METHOD]	访问页面时的请求方法。例如：“GET”、“HEAD”，“POST”，“PUT”
$_SERVER[DOCUMENT_ROOT]	当前运行脚本所在的文档根目录。在APACHE配置文件中定义
$_SERVER[HTTP_REFERER]	链接到当前页面的前一页面的URL地址。不是所有的用户代理（浏览器）都会设置这个变量，而且有的还可以手工修改HTTP_REFERER。因此，这个变量不总是正确真实的
$_SERVER[REMOTE_ADDR]	正在浏览当前页面用户的IP地址
$_COOKIE	通过HTTP cookies传递的变量组成的数组。是自动全局变量
$_GET	通过HTTP GET方法传递的变量组成的数组。是自动全局变量
$_POST	通过HTTP POST方法传递的变量组成的数组。是自动全局变量
$_FILES	通过HTTP POST方法传递的已上传文件项目组成的数组。是自动全局变量
$_REQUEST	此关联数组包含$_GET，$_POST和$_COOKIE中的全部内容
$_SESSION	包含当前脚本中已经注册的session变量的数组
$GLOBALS	由所有已定义全局变量组成的数组。变量名就是该数组的索引

3.4　PHP 中的数组型变量

数组型变量是一组具有相同类型和名称的变量的集合。它是一种很独特的变量，PHP 中的数组可以是一维也可以是多维的，数组内元素的类型也可以是数字、字符甚至是数组变量。

3.4.1　数组变量的初始化

在 PHP 中初始化数组一般有两种方法，一种是同时给数组中所有元素赋值，另一种是单独给数组每个元素赋值。下面通过实例来具体了解这两种方法。

3-5.php：同时给数组所有元素赋值。

```
<html>
<head>
<title>同时给数组所有元素赋值例子</title>
</head>
<body>
<?
$string=array(
"string1",
"string2",
"string3",
"string4",
"string5"
```

```
);                                          //定义一个数组同时给数组所有元素赋值
for($i=0;$i<count($string);$i++)            //循环读取数组内容
{
echo $string[$i];                           //显示数组元素
echo "<br>";                                //输出 HTML 换行符
}
?>
</body>
</html>
```

3-6.php：单独给数组每个元素赋值。

```
<html>
<head>
<title>分别给数组每个元素赋值例子</title>
</head>
<body>
<?
$string[0]="string1";                       //定义数组给数组每个元素单独赋值
$string[1]="string2";
$string[2]="string3";
$string[3]="string4";
$string[4]="string5";
for($i=0;$i<count($string);$i++)            //循环读取数组内容
{
echo $string[$i];                           //显示数组元素
echo "<br>";                                //输出 HTML 换行符
}
?>
</body>
</html>
```

在 PHP 运行环境中分别运行以上两个例子，输出的结果都是一样的，如图 3.5 所示。



图 3.5　给数组赋值执行结果

3.4.2　获取数组中的元素

在给一个数组赋值之后就可以拿来使用了。使用的方法也很简单。只需使用数组加上括号加所需要元素的序号就可以了。需要注意的是，数组下标是从“0”开始的，引用就把需要引用的序号减“1”。如“$String[2]”，就实现了对数组“$String”第 3 个元素的引用。下面能过一个简短的例子来说明，

怎样实现对数组的引用。

```
<html>
<head>
<title>引用数组元素例子</title>
</head>
<body>
<?
$string=array(
"string1",
"string2",
"string3",
"string4",
"string5"
);                                          //定义一个数组
echo "数组的第 3 个元素为：";
echo $string[2];                            //获取数组第 3 个元素
echo "<br>";
echo "数组的第 5 个元素为：";
echo $string[4];                            //获取数组第 5 个元素
echo "<br>";
echo "数组的第 1 个元素为：";
echo $string[0];                            //获取数组第 1 个元素
?>
</body>
</html>
```

保存以上代码为“3-7.php”。在 PHP 运行环境下运行以上代码，执行结果如图 3.6 所示。

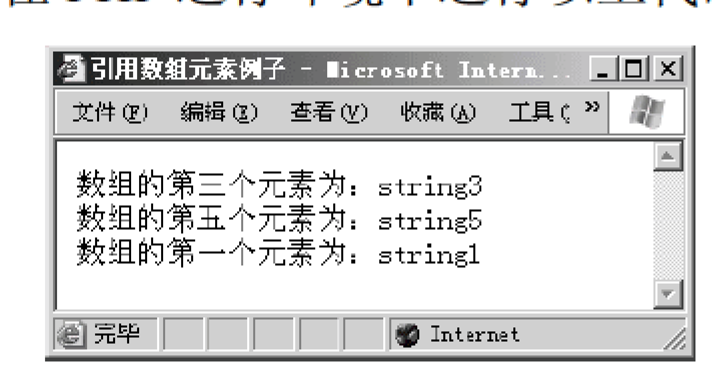


图 3.6　引用数组元素例子输出结果

3.4.3　给数组动态增加元素

一个数组在定义后，它的元素个数并不是一成不变的，程序在运行中可以动态为数组增加元素。要给一个数组动态增加元素，所要做的只是给数组新的元素赋值。下面仍然通过例子来说明这一问题。

```
<html>
<head>
<title>动态给数组增加元素例子</title>
</head>
<body>
<?
$string=array(
```

```
"string1",
"string2",
"string3",
"string4",
"string5"
);                                          //定义一个数组
echo "数组的第三个元素为：";
echo $string[2];                            //获取数组元素
echo "<br>";
echo "数组的第五个元素为：";
echo $string[4];
echo "<br>";
echo "数组的第一个元素为：";
echo $string[0];
echo "<br>";
$string[5]="string6";                       //为数组动态增加元素
$string[6]="string7";                       //为数组动态增加元素
echo "下面的是新增加的数组元素：<br>";
echo "数组的第六个元素为：";
echo $string[5];                            //获取新增加的元素
echo "<br>";
echo "数组的第七个元素为：";
echo $string[6];
?>
</body>
</html>
```

保存以上代码为“3-8.php”。在 PHP 环境中执行以上代码，执行结果如图 3.7 所示。

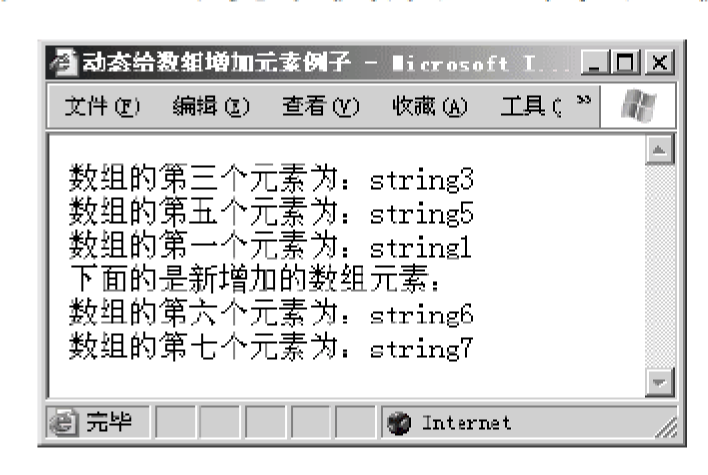


图 3.7　动态给数组增加元素例子执行结果

3.4.4　创建多维数组

一维数组的格式是 Array[]，二维数组的格式就是 Array[][]，多维数组的格式是 Array[][]……[]。和一维数组一样，给多维数组赋值也有两种方法。下面分别通过实例来具体说明。

先来了解一下同时给多维数组所有元素赋值。下面的实例要完成的功能是在定义数组同时，给数组所有元素赋值。

```
<html>
<head>
<title>同时给多维数组所有元素赋值例子</title>
```

```
</head>
<body>
<?
$string=array(
0=>array(
0,
1,
2
),
1=>array(
"string1",
"string2",
"string3",
"string4",
),
2=>array(
"你好！",
"大家好，",
"才是真的好"
)
);                                        //创建二维数组，数组元素也是数组
for($i=0;$i<count($string);$i++)          //通过循环读取外层数组内容
{
for($j=0;$j<count($string[$i]);$j++)      //通过循环读取内层数组内容
    {
        echo $string[$i][$j];             //显示数组元素
        echo ",";
}
echo "<br>";
}
?>
</body>
</html>
```

保存以上代码为“3-9.php”在 PHP 执行环境中执行以上代码，执行结果如图 3.8 所示。

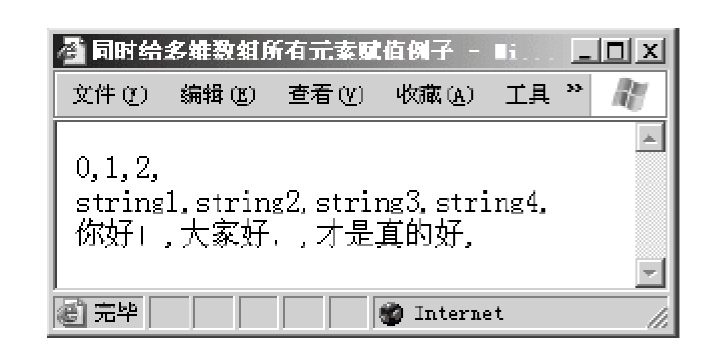


图 3.8　同时给多维数组所有元素赋值例子执行结果

下面再来介绍如何单独给数组每个元素赋值，请参看以下代码：

```
<html>
<head>
<title>单独给多维数组每个元素赋值例子</title>
</head>
<body>
```

```
<?
$string[0][0]=0;                        //单独给多维数组每个元素赋值
$string[0][1]=1;
$string[0][2]=2;
$string[1][0]="string1";
$string[1][1]="string2";
$string[1][2]="string3";
$string[1][3]="string4";
$string[2][0]="你好！";
$string[2][1]="大家好，";
$string[2][2]="才是真的好";
for($i=0;$i<count($string);$i++)        //通过循环读取外层数组内容
{
    for($j=0;$j<count($string[$i]);$j++)    //通过循环读取内层数组内容
    {
        echo $string[$i][$j];           //显示数组元素
        echo ",";
    }
    echo "<br>";
}
?>
</body>
</html>
```

保存以上代码为“3-10.php”在 PHP 运行环境中执行以上代码，执行结果和图 3.8 一样。尽管以上两例数组赋值所采用的方法不同，但都达到了给多维数组赋值的目的。

3.5 本章小结

本章主要介绍了 PHP 中的常量与变量。并详细说明了常量的定义与使用、PHP 中的预定义常量；变量的类型、变量的使用范围、PHP 的系统变量；数组变量的初始化、获取数组中的元素、给数组动态增加元素及多维数组的创建与使用等内容。读者在实际使用时，需要注意以下问题：

- 常量与变量的区别，这一点不同于其他编程语言。
- 引用变量时注意变量名的大小写。
- 数组的引用形式，这也不同于其他语言。

第 4 章

PHP 中的运算符与表达式

- PHP 中的运算符
- 各种运算符的优先级
- PHP 中的表达式
- 运算符与表达式的综合使用

运算符与表达式是 PHP 中十分重要的概念。在 PHP 编程中，表达式是 PHP 程序最重要的基石，而运算符又是构成表达式的基础。可以说，任何复杂的 PHP 程序都是由最基本的运算符和表达式组成的。熟练运用 PHP 中的运算符与表达式，是进行 PHP 编程的基本功。通过本章学习，读者将会对 PHP 中的运算符与表达式有一个全面的认识。

4.1 运　算　符

PHP 中的运算符分为：四则运算符、逻辑运算符、三目运算符和运算符的优先级等。本节将会为读者详细介绍 PHP 中这些运算符的相关知识。

4.1.1 四则运算符

数学运算符号有 4 种，包括："+"（加）、"-"（减）、"*"（乘）、"/"（除）。这些都是人们通常熟悉的符号，PHP 中的运算符基本与此类似，只是多一个"%"求余数的符号。下面通过表 4.1 来具体介绍。

表 4.1　PHP 中的四则运算符

例　子	名　称	结　果
-$a	取反	$a的相反数
$a+$b	加	$a和$b的和
$a-$b	减	$a和$b的差
$a*$b	乘	$a和$b的积
$a/$b	除	$a和$b的商
$a%$b	取余	$a除以$b的余数

在使用"/"符号时，要注意求得的商必定是浮点数，即使能整除也是这样。另外，使用"%"符号进行求余运算时，如果除数$a 为负值则结果也为负值。

4.1.2 逻辑运算符

PHP 中的逻辑运算符有与、或、异或、非 4 种。其中的逻辑与和逻辑或有两种表现形式。下面分别具体说明。

- and，逻辑与。$a and $b 两个表达式求与，只有当两个表达式都为真时返回 True，否则返回 False。
- or，逻辑或。$a or $b 两个表达式求或，当两个表达式都为假时返回 False，否则返回 True。
- xor，逻辑异或。$a xor $b 两个表达式求异或。当两个表达式状态相反时返回 True，否则返回 False。
- !，逻辑非。!$a 一个表达式求非。当$a 为 False 时返回 True，反之返回 False。
- &&，逻辑与。见 and。
- ||，逻辑或。见 or。

之所以"与"与"或"有两种表现形式（逻辑与：and、&&；逻辑或：or、||）是因为它们的运算优先级不同，这一点将在运算符的优先级一节（4.1.4 节）中详细说明。

下面通过实例来说明这些逻辑运算符是怎么用的。由于“&&”等价于“and”，“||”等价于“or”，所以就不再单独说明了。

```
<html>
<head>
<title>逻辑运算符使用实例</title>
<body>
<?
$a=TRUE;                                    //定义逻辑变量真
$b=FALSE;                                   //定义逻辑变量假
if($a and $b) echo "这里为假 1！";          //求与
echo "<br>";
if($a or $b) echo "这里为真 1！";           //求或
echo "<br>";
if($a xor $b) echo "这里为真 2！";          //求异或
echo "<br>";
if(!$a) echo "这里为假 2!";                 //求非
?>
</body>
</html>
```

保存以上代码为“4-1.php”的 PHP 文件。在 PHP 执行环境中执行以上代码，执行结果如图 4.1 所示。从图中可以发现，以上代码中只有两句 ECHO 语句成功运行了。下面分析一下以上程序 4 个项目的运行过程：

- 第 1 个判断对一真一假两项求与。因两者不同时为真，所以返回“假”。
- 第 2 个判断对一真一假两项求或。因二者有一个为真，所以返回“真”。
- 第 3 个判断对一真一假两项求异或。因两者状态不同，所以返回“真”。
- 第 4 个判断对一个真值求非，则返回“假”。

下面把以上代码做如下改动：

if($a and $b)改为 if(!($a and $b))，把 if(!$a)改为 if(!$b)。然后再运行以上代码，运行结果如图 4.2 所示。

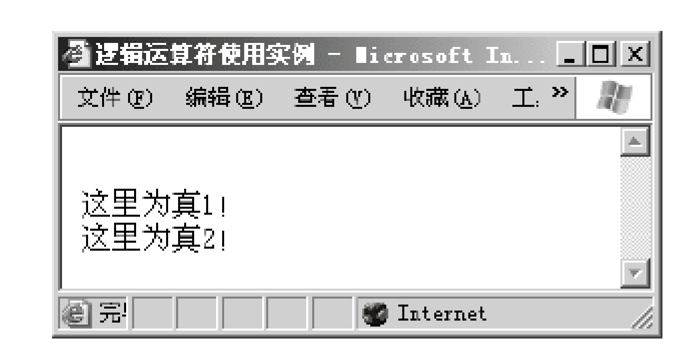



图 4.1　逻辑运算符使用实例执行结果



图 4.2　逻辑运算符使用实例执行结果

从以上实例及改动后的执行情况，可以认识到这些逻辑运算符是如何起作用的。

4.1.3　三目运算符

像 C 语言一样，PHP 中也有三目运算符“?:”。它的运行机制如下：

```
(expr1)?(expr2):(expr3)
```

其中的 Expr1、Expr2 及 Expr3 均为表达式。当表达式 Expr1 为真时则执行后边的 Expr2，反之则执行 Expr3。从分析中不难看出，三目运算符“?:”实际上也就是 if…else 的简化版。

下面通过一个实例来说明三目运算符的使用方法。

```
<html>
<head>
<title>三目运算符使用实例</title>
<body>
<?
$date=Date("D");                                                  //把当前星期赋值给变量
(($date=="Sun")or($date=="Sat"))?($s="周末"):($s="工作");          //判断是否是周末
echo $s;                                                          //显示状态
?>
</body>
</html>
```

保存以上代码为“4-2.php”。在 PHP 执行环境下运行以上代码，如果当前日期的星期为周六或周日时显示“休息”，否则显示“工作”。通过这个例子可以认识到三目运算符的使用方法。

4.1.4　运算符的优先级

在 PHP 中除了前几小节讲到的常用运算符之外，还有其他的一些运算符，那么这么多的运算符，它们的优先级（也就是先执行，后执行的问题）情况是怎么样呢？表 4.2 列出了 PHP 中的运算符，并按顺序给出了它们的优先级。

表 4.2　PHP 中运算符的优先级

<table>
<tr><th>结合方向</th><th>运算符</th><th>附加信息</th></tr>
<tr><td>非结合</td><td>new</td><td>new</td></tr>
<tr><td>左</td><td>[</td><td>array()</td></tr>
<tr><td>非结合</td><td>++ --</td><td>递增/递减运算符</td></tr>
<tr><td>非结合</td><td>! ~ - (int) (float) (string) (array) (object) @</td><td>类型</td></tr>
<tr><td>左</td><td>* / %</td><td>算数运算符</td></tr>
<tr><td>左</td><td>+ - .</td><td>算数运算符和字符串运算符</td></tr>
<tr><td>左</td><td><< >></td><td>位运算符</td></tr>
<tr><td>非结合</td><td>< <= > >=</td><td>比较运算符</td></tr>
<tr><td>非结合</td><td>== != === !==</td><td>比较运算符</td></tr>
<tr><td>左</td><td>&</td><td>位运算符和引用</td></tr>
<tr><td>左</td><td>^</td><td>位运算符</td></tr>
<tr><td>左</td><td>|</td><td>位运算符</td></tr>
<tr><td>左</td><td>&&</td><td>逻辑运算符</td></tr>
<tr><td>左</td><td>||</td><td>逻辑运算符</td></tr>
<tr><td>左</td><td>? :</td><td>三目运算符</td></tr>
</table>

续表

结 合 方 向	运　算　符	附 加 信 息
右	= += -= *= /= .= %= &= |= ^= <<= >>=	赋值运算符
左	and	逻辑运算符
左	xor	逻辑运算符
左	or	逻辑运算符

左联表示表达式从左向右求值，右联相反。从上表中可以发现 PHP 中的运算符有严格的运算优先级。只有搞清楚了它们的优先级，才能正确得出由运算符构成的表达式的值。下面通过实例来说明 PHP 中运算符的优先级在实际中的运用。

```
<html>
<head>
<title>PHP 运算符优先级使用实例</title>
<body>
<?
$a=3*4+5%2;                         //语句 1
echo $a."<br>";
$a =true?0:true?1:2;                //语句 2
echo $a."<br>";
$a=1;
$b=2;
$a-=$b+=3*$b+$a;                    //语句 3
echo $a.",".$b."<br>";
?>
</body>
</html>
```

保存以上代码为“4-3.php”。在 PHP 执行环境中运行以上代码，执行结果如图 4.3 所示。以上结果是怎么得出的？不管多么复杂的表达式，只要按优先级把它分解为简单的表达式，然后就可以分析其结果了。下面对以上 3 个语句逐一作分解。

第 1 个语句相对简单，只用分解为(3*4)+(5%2)即可。3*4 等于 12，5 除以 2 余数为 1。所以就是 12+1 等于 13。

第 2 个语句稍微复杂一点，可以分为两个三目运算符：(true?0:true)?1:2=2。第 1 个三目运算符执行后前面括号内容为“0”，而“0”相当于 False，所以执行第 2 个三目运算符第 2 个表达式。所以就有$a 等于 2。

第 3 个语句就有点复杂了。$a-=$b+=3*$b+$a 分解过的式子等价于这样一组表达式：

$b*3 等于 6，6+$a 等于 7，$b+7 等于 9，$b=9，$a-9 等于-8，$a=-8。

通过分解可以发现输出的结果最后$a=-8、$b=9 是正确。其执行过程如图 4.4 所示。

通过上面的例子，发现只有掌握了 PHP 运算符的优先级，才能把复杂的表达式转化为简单的表达式，从而得出表达式正确的结果。



图 4.3　PHP 运算符优先级使用实例执行结果

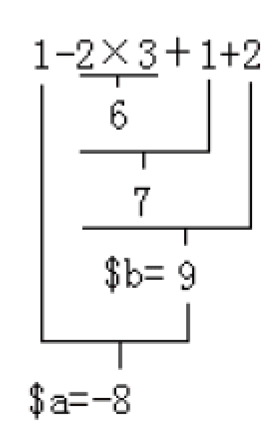


图 4.4　语句 3 执行过程图

4.2　表　达　式

表达式是 PHP 的基石，在 PHP 程序中，几乎所写的任何东西都是一个表达式。何为表达式？可以简单的说，表达式就是“任何有值的东西”。表达式可以为常量、变量或者函数。其中常量和变量是最基本的表达式形式，函数是稍微复杂一点的表达式形式。

例如下面的函数：

```
<?
function foo()
{
    return 5;
}
?>
```

那么输入$c=foo()就相当于写下$c=5。函数也是表达式，表达式的值即为它们的返回值。

4.2.1　表达式中变量的可能值

表达式赋值给一个变量，其中的值可能有 PHP 支持的 4 种标量值（标量值不能拆分为更小的单元，和数组、对象不同）类型：整型值（Integer），浮点数值（Float），字符串值（String）和布尔值（Boolean），还包括两种复合类型（数组和对象）。所以表达式的可能值为 PHP 中所有的变量类型。

4.2.2　赋值表达式的值

一个赋值表达式通常涉及到两个值，如$a=3。这样一个表达式涉及到整型常量 3 的值以及变量$a 的值，它也被更新为 3。还有一个赋值语句本身的值。赋值语句本身求值为被赋的值即 3。因而，这样写“$b=($a=3)”和这样写“$a=3;$b=3;”是一样的。因为赋值操作的顺序是由右到左的，也可以这么写“$b=$a=3”。

通过以上介绍知道，赋值表达式牵涉 3 个值：赋值变量值、被赋的常量值及表达式自身的值。

4.2.3　递增表达式

递增（减）表达式是一种比较特殊的表达式，这类表达式是一个很好的面向表达式的例子。该类表达式有前、后递增和递减。本质上来讲，前递增和后递增均增加了变量的值，并且对于变量的影响是相同的，不同的是递增表达式的值。前递增写作“++$variable”，求增加后的值。后递增写作“$variable++”，求变量未递增之前的原始值。

举一个例子来说明这个问题，代码如下：

```
<html>
<head>
<title>PHP 运算符优先级使用实例</title>
<body>
<?
$a=3;
echo "\$a=3  ";
echo "++\$a 为：";
echo ++$a;                                    //$a 前递增
echo "<p>";
$a=3;                                         //$a 重新赋值为 3
echo "\$a=3  ";
echo "\$a++为：";
echo $a++;                                    //$a 后递增
?>
</body>
</html>
```

保存以上代码为“4-4.php”，在 PHP 执行环境下执行该文件，其执行结果如图 4.5 所示。



图 4.5　递增表达式执行情况

从图 4.5 执行结果可以发现前递增表达式与后递增表达式的值。

4.2.4　比较表达式

一个常用到的表达式类型是比较表达式。这些表达式求值 False 或 True。PHP 支持>（大于），>=（大于等于），==（等于），!=（不等于），<（小于），<=(小于等于)。PHP 还支持全等运算符===（值和类型均相同）和非全等运算符!==（值或者类型不同）。这些比较表达式都是在条件判断语句（如 if 语句）中最常用的，通常被当作条件判断语句的判断条件。

如变量$a 的值为 3，变量$b 的值为 4。则表达式$a<（小于）$b 的值就为真。因为 3 本来就小于 4。

反之$a>（大于）$b 的值就为假。

4.2.5　组合的运算赋值表达式

通过前面对 PHP 表达式的介绍，读者知道如果想要为变量$a 加 1，可以简单的写“$a++”或者“++$a”。但是如果想为变量增加大于 1 的值如 3，其做法是“$a=$a+3”。“$a+3”计算$a 加上 3 的值，并且得到的值重新赋予变量$a，于是$a 的值增加了 3。这个式子还可以用一种更加简短的形式“$a+=3”。这里的意思是“取变量$a 的值加 3，得到的结果再次分配给变量$a”。除了更简略和清楚外，也可以更快地运行。

“$a+=3”的值，如同一个正常赋值操作的值，是赋值后的值。注意它不是 3，而是$a 的值加上 3 后的值（此值将被赋给$a）。任何二元运算符都可以用运算赋值模式，例如“$a-=5”（从变量$a 的值中减去 5），“$b*=7”（变量$b 乘以 7）等。

4.3　运算符与表达式综合运用实例

前面两节介绍了 PHP 中的运算符与表达式的相关知识。这一节，将通过综合的例子来系统的巩固 4.1，4.2 节所讲内容。

```
<html>
<head>
<title>PHP 运算符表达式综合运用实例</title>
<body>
<?
$a="123";
$b=321;
echo $a+$b;                              //字符当数字用
echo "<br>";
echo $a.$b;                              //数字当字符
echo "<br>";
$a=123;
$b=321;
echo $b>$a;                              //比较大小
echo "<br>";
echo $a-23+$b+=$a%3?50:30;               //三目运算符与赋值四则运算
echo "<br>";
$a=123;
$b=321;
$a=$b;                                   //赋值表达式
echo $a==$b;
echo "<br>";
?>
</body>
</html>
```

保存以上代码为“4-5.php”，然后在 PHP 执行环境中执行以上代码，执行结果如图 4.6 所示。

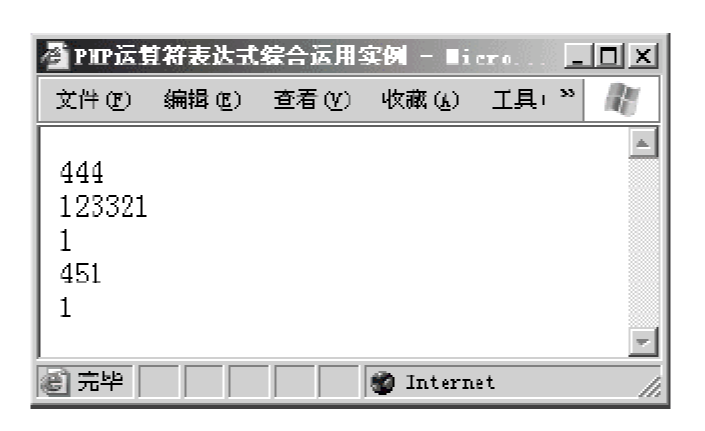


图 4.6　PHP 运算符与表达式综合运用实例

上面例子中的“$a+$b”是实现了把数字字符当作数字使用，两数字相加值为 444。第 2 句则是实现了把数字当字符用，两字符相加为 123321。关键就在于使用者用什么符号。如果用“+”，就把字符当数字；如果用“.”就把数字当字符。

第 3 句是实现对两个数字比较大小。因为 321 大于 123，所以返回真值即“1”。

第 4 句是三目运算符与赋值运算符及四则运算的混合使用。$a-23+$b+=$a%3?50:30;因为数学运算符遵循从右到左的标准，而三目运算符则是从左向右。所以这个表达式可以分为以下几个子过程。$a%3 值为“0”，执行三目运算符“:”后的内容，返回 30，$b=$b+30 等于 351，$a−23=100，100+351 等于 451。

第 5 句先定义两个不同的变量，再通过赋值操作把变量$b 的值赋给变量$a，然后判断二者是否相等，在经过赋值操作后，二者的值当然相等，所以比较结果返回真即“1”。

4.4　本章小结

这一章主要学习了 PHP 中的运算符与表达式。内容包括 PHP 的四则运算符、逻辑运算符、三目运算符、运算符的优先级以及 PHP 中的表达式。它们的重要性不言而喻。正如本章开头所说的那样——运算符与表达式是 PHP 最重要的基石。所以，只有掌握了这些内容，才能谈及 PHP 编程。在实际运用运算符及表达式时要注意以下几个问题：

- ❑ 运算符的优先级。如果设置不合理的优先级，得出的就不会是预想的结果。
- ❑ 递增表达式的值。从 4-4.php 的执行结果可以了解到，前递增与后递增表达式有不同的值，这一点需要引起注意。
- ❑ 复杂的组合运算赋值表达式的值。表达式的值为变量在经过运算之后的值。

第 5 章

PHP 中的流程控制

- PHP 中的 if 判断
- PHP 中的 switch…case 判断
- PHP 中的 while 循环
- PHP 中的 for 循环
- 流程控制综合运用实例

无论在何种编程语言中，流程控制都是很基础而且很重要的内容。由于 PHP 的大部分语法都是继承了 C 语言的特点。因此在流程控制方面，PHP 有着和 C 语言类似的流程控制。PHP 的流程不像 ASP 那样可以使用 goto 的 BASIC 语法。本章将介绍 PHP 中的判断与循环语句，包括 if…else 判断、switch…case 多重判断、while 循环、do…while 循环、for 循环等内容。通过本章的学习将使读者掌握 PHP 流程控制知识，为编写大型程序，奠定坚实的基础。

5.1　if…else 判断

if…else 判断，是流程控制中判断执行的一种。该语句执行时先对某条件进行判断，然后根据判断结果做出相应的操作。它又可以详细地分为 3 种：简单的 if 判断、if…else 判断和 if…else…else 判断。本节具体讲解 if…else 判断。

5.1.1　简单的 if 判断

if 判断是流程控制中最简单的一种。只判断某条件是否为真，如果为真就执行特定的语句。例如下面这种情况：

```
if(expr)
{
statement
}
```

如果在执行的 statement（语句）多于一句，就要使用“{}”把它们括起来，表示一个区段。如果要执行的语句只有一句，就可以省略大括号标记。下面通过一个简单的例子来说明如何使用 if 判断。

```
<html>
<head>
<title>IF 判断使用实例</title>
</head>
<body>
<?
if($cost>100)                              //如果商品的价格超过 100 元执行操作
echo "太贵了，不买！";
?>
</body>
</html>
```

保存以上代码为“5-1.php”。在 PHP 运行环境中执行以上语句。执行结果是什么？什么也没发生。这是因为并没有定义一个变量$cost，没有定义的变量就是空值，作为数值就是“0”。而“0”是小于 100 的，这明显不符合 if 后面的判断内容（$cost>100），所以什么也不执行。这里在 if 判断之前加入如下一句：

```
$cost=101;
```

再执行以上代码，结果将会如图 5.1 所示。

由于此时满足$cost>100 的条件，所以 if 后面的内容会被正确执行。从这个例子中能够认识到 if 判断语句是怎么起作用的。



图 5.1　IF 判断使用实例输出结果

5.1.2　if...else 判断

if 判断只对判断结果为真的情况执行操作，这在很多情况下是不够的。于是就有了 if…else 这种形式的判断。与 if 判断不同，if…else 不仅对判断结果为真的情况执行操作，对非真的情况也执行相应的操作。

为了简单说明问题，继续使用 5.1.1 小节的例子，只是把 5.1.1 小节的例子做简单的修改。具体代码如下：

```
<html>
<head>
<title>IF…ELSE 判断使用实例</title>
</head>
<body>
<?
$cost=101;
if($cost>100)                          //如果商品的价格超过 100 元执行操作
echo "太贵了，不买！";
else
echo "还挺便宜的，就买它吧！";         //如果价格不超过 100 元的操作
?>
</body>
</html>
```

保存以上代码为“5-2.php”。在 PHP 运行环境下执行以上代码，执行结果和图 5.1 是完全一样的。因为$cost 的值大于 100，所以 if 语句结果为真，跟在后面的 echo 语句就执行了，即显示出不买的信息。

把上例中的

```
$cost=101;
```

改为：

```
$cost=99;
```

重新执行以上代码，执行结果则会如图 5.2 所示。

因为，经过改动$cost 小于 100，if 语句不再起作用。而 else 语句则起了作用。所以就会打印出购买商品的信息。

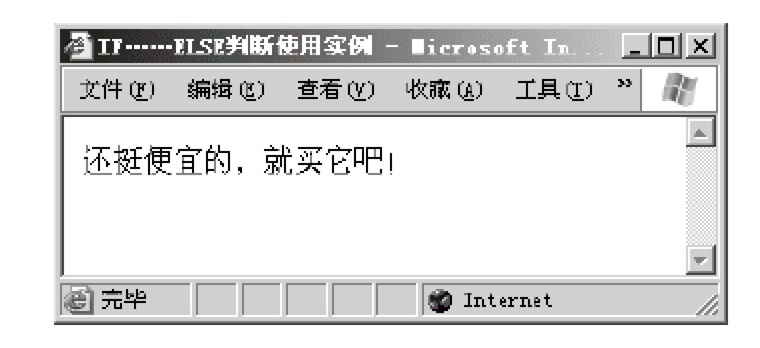


图 5.2　IF…ELSE 判断使用实例执行结果

5.1.3　if…else…else 多重判断

虽然 if…else 的判断比单纯的 if 语句多了一重判断，但现实情况可能还要复杂，要判断的情况会超过两种。如判断学生成绩，不能只判断及格或不及格，而要判断优、良、中、差。这时不管是简单的 if 判断，还是 if…else 判断都已经不能胜任这项工作了。这时就要用到 if…else…else 的多重判断。

if…else…else 多重判断的使用方法如下：

```
if(expr)
{
statement
}
elseif(expr)
{
statement
}
……
elseif(expr)
{
statement
}
else
{
statement
}
```

它的运行机制是：先进行一次判断，如果为真就执行语句并跳出；否则再进行判断，如果为真就执行语句跳出；否则再进行判断……直到所有判断结束，最后剩余的情况，再执行相应语句结束。

下面通过一个实例来说明怎样使用 if...else...else 多重判断。该代码实现根据学生的成绩，来判断其优、良、中、差的级别。

```
<html>
<head>
<title>IF……ELSEIF……ELSE 多重判断使用实例</title>
</head>
<body>
<?
$score=99;
if($mark>90)                        //如果成绩大于 90 分执行操作
echo "成绩级别为：优！";
elseif($score >70)                  //如果成绩在 70~90 之间执行操作
echo "成绩级别为：良！";
elseif($score >60)
echo "成绩级别为：中";
else                                //如果成绩低于 60 执行操作
echo "成绩级别为：差！";
?>
</body>
</html>
```

保存以上代码为 PHP 文件“5-3.php”，然后在 PHP 执行环境中执行该文件。通过改变$score 的值可以看到当值大于 90 时返回“优”，当值在 70~ 90 之间返回“良”，当值在 60~70 之间返回“中”，当值在 60 以下时返回“差”。通过该例子，能够清楚地认识到 IF...ELSEIF...ELSE 多重判断是如何运行的。

5.2　switch…case 多重判断

多重判断除了 if…else…else 之外，还有另外一种就是 switch…case。与 if 多重判断相比较，switch…case 更简洁明了。switch…case 的运行机制如下：

```
switch (expr) {
  case expr1:
    statement1;
    break;
  case expr2:
    statement2;
    break;
    :
    :
  default:
    statementN;
    break;
}
```

expr 通常是变量名称，case 后的 exprN 通常为变量的值，statementN 为符合该值时执行的语句。default 为除以上所有判断之外的情况，最后使用 break 跳出过程。

下面通过一个实例来实际使用 switch…case。代码如下：

```
<html>
<head>
<title>SWITCH……CASE 多重判断使用实例</title>
</head>
<body>
<?
switch (date("D")) {                          //当前星期作为判断条件
  case "Mon":                                 //星期一的情况
    echo "星期一";
    break;
  case "Tue":                                 //星期二的情况
    echo "星期二";
    break;
  case "Wed":                                 //星期三的情况
    echo "星期三";
    break;
  case "Thu":                                 //星期四的情况
    echo "星期四";
    break;
  case "Fri":                                 //星期五的情况
    echo "星期五";
    break;
  default:                                    //除以上之外的其他情况
```

```
        echo "过周末";
        break;
}?>
</body>
</html>
```

把以上代码保存为 PHP 文件“5-4.php”，然后在 PHP 运行环境下执行，就会返回中文的星期。这个例子如果用 if 来判断实现就会比较复杂。从这个例子能够看出 if 判断与 switch…case 的不同。

在使用 switch…case 语句的时候，要把出现机率大的情况放到最前边，这样可以提高程序执行效率。因为在进行完一次判断，如果符合条件，执行完相关语句就跳出整个过程了。如果把出现机率大的情况放到最后，一旦出现，就得执行完所有判断，这样效率就低得多。

5.3 while 循环

5.1、5.2 节介绍了 PHP 流程控制中的判断。从本节开始为读者介绍循环。循环也是很重要的一类流程控制，通过循环执行某些特定语句要比多次执行效率高。PHP 中的循环共有 while 循环、do…while 循环和 for 循环三大类。这些都将在以后的几节中陆续为读者介绍。这一节先来学习 while 循环。

5.3.1 单纯 while 判断循环

在循环控制中，有一种最简单的循环模式，即先判断条件是否符合特定要求，如果符合就执行特定操作，然后再判断，直到条件不符合则退出循环。但如果一开始条件就不符合要求，则一次也不执行，直接跳出循环。这种循环模式在 PHP 中的表现就是 while 循环。while 循环的使用模式如下：

```
while (expr)
{
statement
}
```

其中的 expr 即为特定的条件，statement 则为执行的操作。下面通过一个简单的例子来说明 While 循环的使用方法。代码如下：

```
<html>
<head>
<title>WHILE 循环使用实例</title>
</head>
<body>
<?
$i=1;                                   //初始化变量
while($i<10)                            //判断变量是否小于 10
{
     echo "第：".$i."次循环";            //执行操作 1
     echo "<br>";                       //执行操作 2
     $i++;                              //变量自增
```

```
}
?>
</body>
</html>
```

保存以上代码为“5-5.php”。在 PHP 运行环境上执行以上代码，结果如图 5.3 所示。



图 5.3　WHILE 循环使用实例执行结果

程序先初始化一个变量$i=1，然后再判断变量是否小于 10，如果小于 10 就执行大括号内的操作。当然，现在的变量小于 10，于是就打印出“第：1 次循环”，并自增 1，现在变量值就为 2。以此类推当循环执行到第 10 次，变量值就为 10，而不符合条件了，于是就跳出循环，结束整个过程。

在使用 while 循环时要注意一个问题，即必须在 while 执行体中使判断条件的对象有所改变。如果没有的话就成为死循环，而永无止境了。上例中如果不让变量自增，则变量的值永远等于 1，循环就会无休止地执行下去，直到耗尽系统资源。

5.3.2　使用 break 跳出循环

在使用 while 循环时，有时并不需要执行到满足 while 要求的条件。在循环执行过程中就可以对执行情况进行判断，如果满足某一条件就使用 break 跳出循环。它的实现过程如下：

```
while (expr)
{
statement
if(expr1)
{
	break;
}
}
```

break 的作用就是跳出当前循环。下面举一个例子来说明，还继续使用 5.3.1 小节的代码，但要对其做一些小小的改动。如下：

```
<html>
<head>
<title>使用 BREAK 跳出循环实例</title>
</head>
<body>
<?
$i=1;                                   //初始化变量
while($i<10)                            //判断变量是否小于 10
{
	echo "第：".$i."次循环";              //执行操作 1
	echo "<br>";                          //执行操作 2
```

```
        $i++;                               //变量自增
        if($i==9)                           //判断变量情况
        break;                              //如果满足条件就跳出循环
}
?>
</body>
</html>
```

保存以上代码为“5-6.php”，查看以上代码与 5.3.1 小节代码 5-5.php 的不同，多出了这样两句。

```
if($i==9)
break;
```

执行结果会有什么不同呢？在 PHP 运行环境中执行以上代码，会得到如图 5.4 所示的结果。

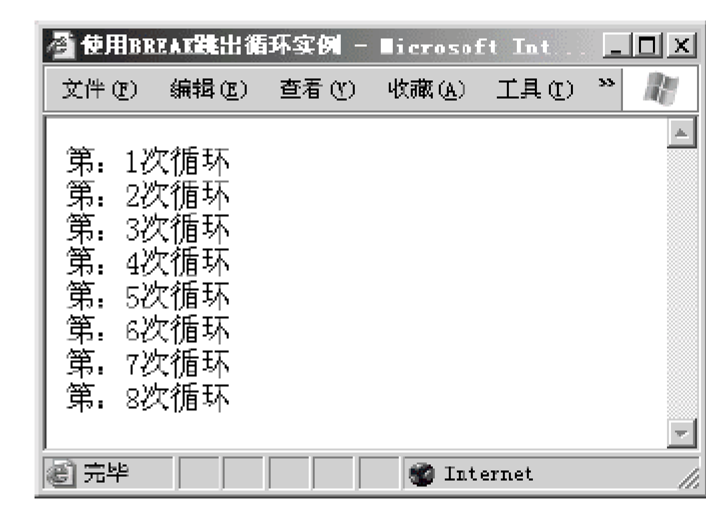


图 5.4　使用 BREAK 跳出循环实例执行结果

可以发现，与 5.3.1 小节的例子相比，少执行了一次。如果没有 if($i==9) break;这样两句，程序一直会执行到$i<10，即当$i 等于 9 时依然是满足条件的。可当使用了没有 if($i==9) break; 后，循环就在$i 等于 9 时跳出了，不再执行。所以看不到第 9 次循环。通过这个例子就可以看到 break 是如何起作用的。

另外，break 除了可以运用到 while 循环中之外，还能运用到 for 循环以及前面提到的 switch…case 多重判断中。作用都是跳出当前运行的过程。

5.3.3　使用 continue 语句

在循环结构中，continue 用来跳过本次循环中剩余的代码，并在条件求值为真时，无条件执行下一次循环。

下面通过一个实例来说明 continue 的工作原理。代码仍然采用前几节的代码，只是做少许改动，这样能更明白地认识 continue 语句是如何起作用的。代码如下：

```
<html>
<head>
<title>使用 CONTINUE 跳出本次循环实例</title>
</head>
<body>
<?
$i=0;                                       //初始化变量
while($i<10)                                //判断变量是否小于 10
{
        $i++;                               //变量自增
        if($i==5)                           //判断变量情况是否为 5
        {
                continue;                   //如果满足条件就跳出本次循环
        }
```

```
        echo "第：".$i."次循环";                  //执行操作 1
        echo "<br>";                              //执行操作 2
}
?>
</body>
</html>
```

保存以上代码为“5-7.php”。在 PHP 执行环境下运行以上代码，执行结果如图 5.5 所示。

从图 5.5 中可以看到，由于 continue 的存在，第 5 次循环并没有被执行。

和 break 一样，continue 同样也可以用在 while、for、switch 之中，其作用原理都是一样的。另外，在使用 continue 时还要注意的一点是，continue 后面的“;”是不能省略的。如果省掉会导致不希望的结果出现。

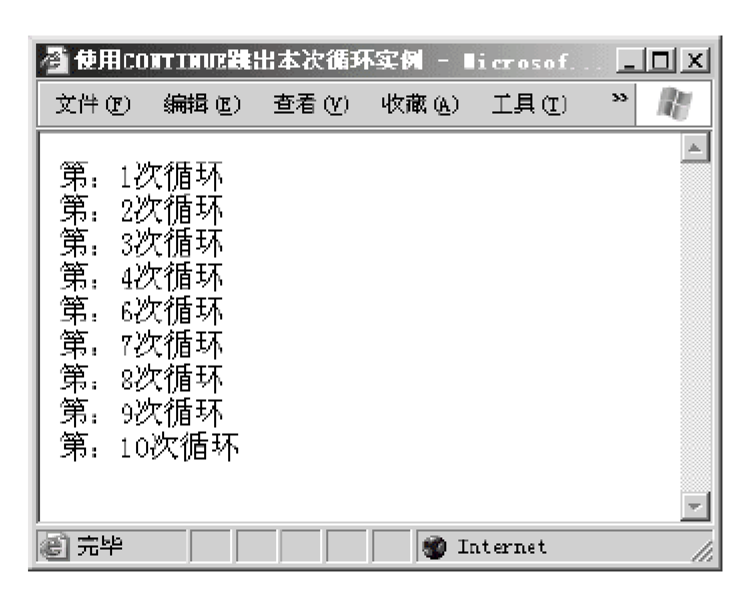


图 5.5　使用 CONTINUE 跳出本次循环实例执行结果

5.4　do…while 循环

这一节来介绍 do…while 循环。do…while 循环与单纯的 while 循环不同。单纯的 while 循环是先判断条件是否为真，如果为真则执行，否则就不执行循环。也就是说，while 循环有可能一次也不执行（初始条件是非真的情况）。而 do…while 循环则与此不同，它是先执行一次循环，然后再判断条件是否为真，如果为真继续执行，否则就跳出循环。它的执行模式如下：

```
do
{
  statement
}while (expr);
```

下面通过一个实例来说明 do…while 是如何起作用的。还是使用前几节的代码，只是作了少许改动，这样可以更清楚地了解到它们之间的异同。代码如下：

```
<html>
<head>
<title>DO……WHILE 循环使用实例</title>
</head>
<body>
<?
$i=1;                                             //初始化变量
do                                                //开始执行循环
{
        echo "第：".$i."次循环";                  //执行操作 1
        echo "<br>";                              //执行操作 2
        $i++;                                     //变量自增
}
```

```
while($i<1)                          //判断变量是否小于 1
?>
</body>
</html>
```

保存以上代码为“5-8.php”。在 PHP 运行环境中执行以上代码，运行结果如图 5.6 所示。



图 5.6　DO...WHILE 循环使用实例输出结果

如果要是在 while 循环中，循环肯定一次也不执行就退出了。因为变量$i 的值等于 1，不满足$i<1 的条件。但在 do…while 循环中则不然，循环先执行一次再判断，由于已经不满足条件于是就退出循环。

5.5　for 循环

for 循环是 PHP 中最复杂的循环结构。它的行为和 C 语言相似。for 循环的执行模式如下：

```
for (expr1; expr2; expr3)
{
    statement
}
```

- 第一个表达式 expr1 在循环开始前无条件求值一次。
- expr2 在每次循环开始前求值。如果值为 True，则继续循环，执行 statement 语句。如果值为 False，则终止循环。
- expr3 在每次循环后被执行。

每个表达式都可以为空。expr2 为空意味着将无限循环下去。如果 expr2 为空，循环无限执行，任何使用者都不想出现这种结果，但用户也可以在循环体里加入 break 语句。当某个条件为真时，就执行 break 语句来跳出 for 循环。

下面通过一个实例来实现 for 循环。代码如下：

```
<html>
<head>
<title>FOR 循环使用实例</title>
</head>
<body>
<?
for($i=1;$i<10;$i++)                          //FOR 循环开始
{
     echo "第：".$i."次循环";                  //要执行的语句
}
?>
</body>
</html>
```

保存以上代码为“5-9.php”。在 PHP 执行环境中运行，执行结果与图 5.3 完全一样（除了 HTML 显示的标题）。

通过 5.3 节对 while 循环的介绍，对比本节对 for 的介绍，可以发现两者的不同。在实际应用中，若循环有初始值，并且都需要累加（或者累减），这时使用 for 循环就比使用 while 循环要好。

5.6　流程控制综合运用实例

这一节将通过一个实例，把判断与循环两种方法结合起来，做一个实际应用的例子。通过这个例子，巩固本章所学的内容。

以下例子实现的功能是根据二维数组的内容，以表格的形式分类打印出数组的全部内容，并以不同的背景颜色显示大类别及小类别。本例代码如下：

```
<html>
<head>
<title>流程控制综合运用实例</title>
<head>
<body>
<?
//首先定义一个数组——图书类型数组
$type[0][0]="学生用书";                    //第 1 个大类别
$type[0][1]="学生教材";                    //第 1 大类中的第 1 小类
$type[0][2]="教辅用书";
$type[0][3]="课外读物";
$type[0][4]="考试题集";
$type[1][0]="名著";                        //第 2 个大类别
$type[1][1]="中国古典";                    //第 2 大类中的第 1 小类
$type[1][2]="世界名著";
$type[1][3]="英文原著";
$type[2][0]="考试用书";                    //第 3 个大类别
$type[2][1]="公务员";                      //第 3 大类中的第 1 小类
$type[2][2]="会计师";
$type[2][3]="医药师";
$type[3][0]="儿童读物";                    //第 4 个大类别
$type[3][1]="看图识字";                    //第 4 大类中的第 1 小类
$type[3][2]="动漫人物";
$type[4][0]="武侠小说";                    //第 5 个大类别
$type[4][1]="金庸小说";                    //第 5 大类中的第 1 小类
$type[4][2]="古龙小说";
$type[4][3]="玄幻小说";
echo "<table border=\"1\">";               //打印表格头
for($i=0;$i<count($type);$i++)             //外层循环
{
      $s=0;                                //定义循环标识变量
      for($j=0;$j<count($type[$i]);$j++)   //内层循环
      {
```

```
            if($s%2==0) echo "<tr>";                        //如果标识为偶数新起一行
            $s++;                                           //标识自增
            if($j==0)                                       //判断是否为大类别
            echo "<td colspan=2 bgcolor=\"#cccc00\">";      //打印大类别的表格
            else echo "<td bgcolor=\"#ccccff\">";           //打印小类别的表格
            echo $type[$i][$j];                             //输出数据
            echo "</td>";                                   //表格结束
            if($j==0)                                       //判断是否为大类别
            {
                $s++;                                       //如果为大类别则标识再次自增
            }
            if($s%2==0) echo "</tr>";                       //如果大类别一格或小类别两格则表格的地行结束
            if($s==(count($type[$i])+1) && count($type[$i])%2==0)//判断小类别项为奇数的情况
            echo "<td bgcolor=\"#ccccff\"> </td></tr>";//在后面添加空表格
        }
}?>
</body>
</html>
```

把以上代码保存为 PHP 文件“5-10.php”。在 PHP 运行环境中执行以上代码，执行结果如图 5.7 所示。

这个例子中的流程控制比较复杂，要考虑多种情况。如大类别只占一格的情况、小类别占两格的情况、小类别结尾不足两格的情况等。通过这个例子，把本章学到的判断、循环的知识都运用到里面了。同时也使读者了解到，在使用 PHP 编程时，如何使用流程控制来处理复杂的问题。

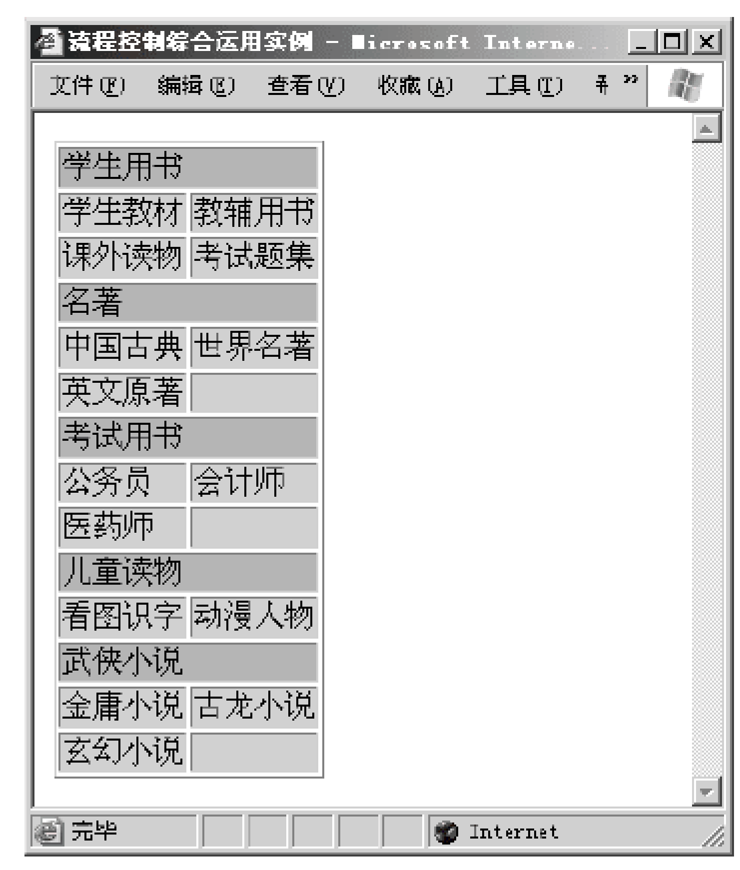


图 5.7　流程控制综合运用实例执行结果

5.7　本 章 小 结

流程控制是 PHP 编程的基础，很多算法的设计都要考虑到流程控制。这一章学习了 PHP 中的流程控制，具体内容包括 if 判断、switch...case 多重判断、while 循环、do...while 循环、for 循环等内容。通过本章的学习使读者掌握 PHP 流程控制知识，为编写大型程序奠定坚实的基础。

第 6 章

PHP 中的函数

- 什么是函数
- 函数的参数
- 函数的返回值
- PHP 中系统函数的使用
- 使用自定义函数

函数是 PHP 最重要的组成部分。如果说前几章介绍的变量、表达式、流程控制是 PHP 的基础的话，那么可以说函数就是 PHP 的主体。什么是函数呢？简单的说，函数就是为了完成特定功能而作为一个整体存在的代码块。PHP 中有大量的库函数，同时也允许用户自定义函数。本章就带领读者来认识一下函数。本章内容包括：什么是函数、函数的参数、函数的返回值、PHP 内部函数的使用、PHP 加载外部函数、如何自定义函数等。

6.1　什么是函数

简单的说，函数是为了完成特定功能，而作为一个整体存在的代码块。例如，求绝对值函数 abs()完成的功能是求一个数的绝对值，而且它也是独立存在的，并不受其他变量或函数的影响。函数采用以下方法来定义。

```
function f_name($arg)
{
      expr;
      return $retval;
}
```

以上代码中 f_name 为函数名，以名字来区别其他函数。$arg 为函数的参数，参数是在函数执行中要传递的值，它跟在函数后面的括号里。如果要求“-3”的绝对值就要用参数来传递“-3”，如 abs(-3)。expr 为函数执行的语句，$revel 为函数的返回值，返回值并不是每个函数都有。返回值指完成函数后返回到主程序中的值。如 abs(-3)的返回值为“-3”的绝对值“3”。其中函数名的命名规则与 PHP 中的变量命名规则相同。有效的函数名以字母或下划线打头，后面跟字母，数字或下划线。

在 PHP 3 中一个函数在调用之前必须已经被初始化。如果调用一个未被定义的函数，将会导致错误。从 PHP 4 开始，就不再有这种限制，这意味着可以先调用一个并未被定义的函数，然后再去定义函数。但是如果函数的定义是有条件的，那么在这有条件的定义发生前，是不能被调用的。就像下面这两种情况。

1．有条件的定义

具体内容请参看以下代码：

```
<?
$makefoo=true;
/*不能在这里调用函数 foo
因为它现在并不存在
但是可以调用函数 bar*/
bar();                                                              //调用函数 bar
if($makefoo)                                                        //定义函数据条件
{
      function foo()                                                //定义函数 foo
      {
            echo "I don't exist until program execution reaches me.\n";    //输出字符串
      }
}
/*现在可以调用函数 foo 了，
因为它已经被定义*/
if ($makefoo) foo();                                                //调用函数 foo
function bar()                                                      //无条件定义函数 bar
{
      echo "I exist immediately upon program start.\n";
```

```
}
?>
```

以上为第 1 种情况，虽然定义函数的条件为真，但是在有条件的定义前，函数是不能被调用的。即如果某条件为真则再去考虑是否去做某件事，如果这条件根本不存在，就不考虑去做这件事，更不用说怎么去做了。

这里举一个形象的例子。例如，一个家长叫小孩子去买东西。小孩子如何去买东西可以看作是一个函数。但是小孩也给家长提了条件，如果给 1 元辛苦费才去买。即只有满足了给 1 元钱的条件，才去执行买东西这个函数，也才去考虑怎么去买。如果条件根本不存在，那么函数就相当于没有定义，当然也不会执行了。

2．函数的嵌套定义

第 2 种情况是在某一函数体中定义另一个函数。只有当外层函数被调用时，内层函数才会被正确定义。所以也只有在调用外层函数后，才能调用内层函数。示例如下：

```
<?
function foo()
{
	function bar()
	{
		echo "I don't exist until foo() is called.\n";
	}
}
/*现在不能调用函数 bar
因为它还未被定义*/
foo();
/*现在可以调用函数 bar，
因随着函数 foo 的调用
函数 bar 也被正确定义了*/
bar();
?>
```

这种情况也很容易理解，与第 1 种情况存在某些相似之处。只是把定义函数的条件转化为了某一函数的执行。

如果不把上面比方的例子中的家长给钱看成一个条件，而是一个函数，就变成了第 2 种函数的嵌套定义了。即家长给钱的函数执行了，那么孩子去买东西这个函数就会被定义，然后就可以调用函数，如怎么去买东西了。

另外在使用函数时还有一点需要注意。与变量不同，函数是大小写不敏感的。如定义的函数为 bar()，实际使用时完全可以通过 bar()来调用它。但通常情况下，为了避免混淆，调用时还是使用定义时的名字。

6.2　函数的参数

在 6.1 节介绍函数的定义中，提到了函数的参数：

```
function f_name($arg)
{
	expr;
	return $retval;
}
```

以上代码中的$arg 就是一个参数。那么究竟什么是函数的参数？

PHP 中的函数按有无参数可分为：有参数函数和无参数函数两种。函数参数只存在于有参数的函数之中。函数参数就是函数名称后圆括号内的常量值、变量、表达式或函数。当定义函数时，这时的参数因为无实际值，所以称其为形式参数，形式参数不能是常量值。当调用该函数时，这时的参数有实际的值称为实际参数。形式参数的类型说明在函数名后的“()”之内。

下面通过一个简单的例子来说明函数参数的使用。

```
<html>
<head>
<title>函数参数的使用实例</title>
</head>
<body>
<?
function B_I_text($text)                    //定义有参数函数
{
	echo "<b><i>".$text."</i></b>";             //打印字符并加入粗体斜体效果
}
$string="PHP 编程是一件很简单的事情";         //定义变量
echo $string;                               //打印变量
echo "<br>";
B_I_text($string);                          //用实际参数调用函数
?>
</body>
</html>
```

保存以上代码为“6-1.php”。在 PHP 执行环境中运行该代码，执行结果如图 6.1 所示。

上例中，先定义了有形式参数的函数。然后定义变量，先显示变量，再通过实际参数调用函数。从这个例子中，读者可以了解到函数的参数是如何传递值到函数体的。



图 6.1　函数参数的使用实例执行结果

在使用函数参数时还应该注意一个问题。函数的参数是有类型限制的，即某一函数的参数可能只对应某一种类型。如果参数的类型与函数要求的类型不一致，就会发生错误，可能会返回用户不希望的结果。如 abs()函数的作用是计算一个数的绝对值，所以它的参数只能为整型或者浮点型数。如果用字符或者数组作为函数的参数，则一定会出现用户不希望的结果。

另外，有的函数有默认的值的参数。这时的参数就变成了可选参数，即调用该函数时可以不加有默认值的参数。函数将用默认值来替换该参数。下面举一个例子来说明：

```
<html>
<head>
<title>函数有默认值参数的使用实例</title>
```

```
</head>
<body>
<?
function B_I_text($text,$color="#000000")

//定义有参数函数，其中$color 参数有默认值
{
    echo "<font color=".$color.">";

//使用颜色参数
    echo "<b><i>".$text."</i></b>";

//打印字符并加入粗体斜体效果
    echo "</font>";
}
$string="PHP 编程是一件很简单的事情";                //定义变量
echo $string;                                        //打印变量
echo "<br>";
B_I_text($string);                                   //调用函数，无$color 参数
Echo "<p>";
B_i_text($string,"red");                             //调用函数，加入$color 参数
?>
</body>
</html>
```

保存以上代码为“6-2.php”。在 PHP 执行环境中运行该文件，执行结果如图 6.2 所示。

上例先定义了一个有两个参数的函数，其中的$color 参数有默认值。即当调用函数时，如果不加可选参数$color，将使用默认值“#000000”，即用黑色打印字体。所以第 1 次调用该函数时没有使用$color 函数，打印出的字体是黑色的。第 2 次调用时使用了“red”作为$color 参数，则打印出的字体就是红色的。

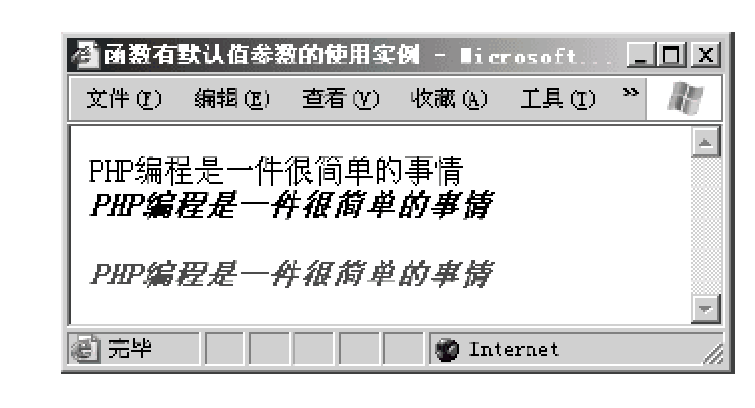


图 6.2　函数有默认值参数的使用实例执行结果

6.3　函数的返回值

在 6.1 节讲到函数时，同时也提到了函数的返回值。

```
function f_name($arg)
{
    expr;
    return $retval;
}
```

即以上代码中的$retval。函数通过 return 来返回值。函数的返回值可以是数值、字符等变量。下面通过一个实例来说明函数的返回值是如何使用的。

```
<html>
<head>
<title>函数返回值的使用实例</title>
</head>
<body>
<?
function cube($num)                                    //定义有参数函数
{
    return $num*$num*$num;                             //将参数连乘三次的值作为返回值
}
$i=3;
echo $i."的三次方为：".cube($i);                       //有实际参数调用函数
?>
</body>
</html>
```

保存以上代码为“6-3.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.3 所示。



图 6.3　函数返回值使用实例执行结果

另外，函数不能有多个返回值，但是为了返回多个值，可以将数组作为一个函数的返回值，从而实现多个值的返回。下面通过一个实例来说明，函数是如何将一个数组作为其返回值的。

```
<html>
<head>
<title>函数将数组返回值的使用实例</title>
</head>
<body>
<?
function E_num($num1,$num2)                            //定义函数有两个参数
{
    if($num1>$num2)                                    //如果前面数大两者互换
    {
        $temp=$num1;
        $num1=$num2;
        $num2=$temp;
    }
    for($i=$num1;$i<$num2;$i++)                        //循环比较两数之间的值
    {
        if($i%2==0)                                    //选出其中的偶数
        {
            $t[$j]=$i;                                 //把结果赋值给数组元素
            $j++;
        }
    }
    return $t;                                         //把数组$t 作为函数返回值
}
$a=3;                                                  //定义变量
```

```
$b=20;
$c=E_num($a,$b);                                    //调用函数
echo $a."到".$b."之间的偶数为：";
for($i=0;$i<count($c);$i++)                          //遍历数组
{
      echo $c[$i];                                  //显示结果
      echo "<br>";
}
?>
</body>
</html>
```

上面代码定义了一个函数 E_num(int num1,int num2)，它的作用是求出两个数之间所有的偶数。因为两数之间的偶数可能为多个数，所以要返回多个数值。这就要在函数中把这些数值定义到数组中，然后把数组当作函数的返回值返回。

把以上代码保存为“6-4.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.4 所示。

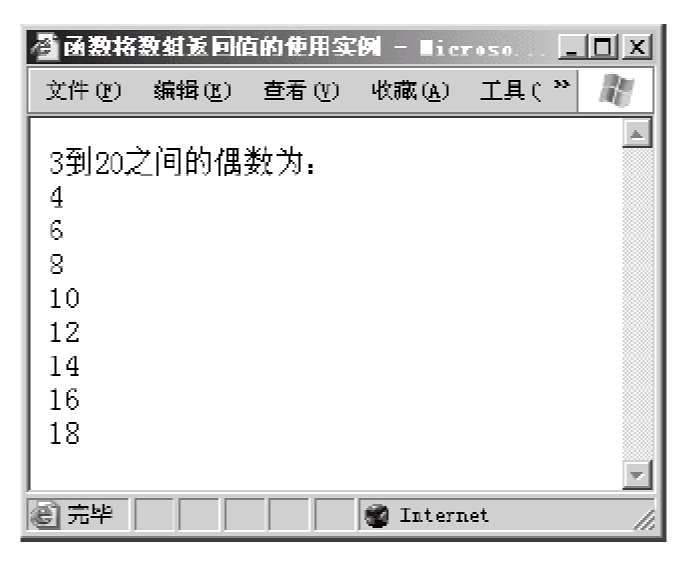


图 6.4　函数将数组返回值的使用实例执行结果

函数的返回值部分，这一节就介绍到这里。相信通过上面的几个例子，对函数的返回值的讲解，读者会有一个深刻的印象。

6.4　PHP 内部函数的使用

PHP 为用户提供了丰富的库函数即内部函数，能否熟练的使用 PHP 的内部函数，是衡量一个 PHP 程序员合格与否的标准。那么如何使用 PHP 中的库函数呢？由于内部函数是集成在 PHP 解释器中的，所以它不用被定义，可以直接使用。使用时只是要注意函数的参数类型、调用方法、返回值及格式即可。

相对于用户自定义函数来说，PHP 的内部函数使用更简单。一是因为它不用定义；二是不用担心函数体本身会出错。如果在使用库函数时返回了不希望出现的结果，那么不要怀疑是系统函数有问题。因为这些函数都是经过了反复的实践证明的，是绝对正确无误的。

下面就通过两个例子，来说明在 PHP 中如何使用库函数。

调用常用的数学函数。

```
<html>
<head>
<title>PHP 库函数的使用实例 1</title>
</head>
<body>
<?
echo abs(-3);                          //调用求绝对值的库函数
echo "<p>";
echo floor(5.321);                     //调用取整函数
echo "<p>";
```

```
echo pi();                              //调用圆周率函数
echo "<p>";
echo min(5,1,3,7,8);                    //调用求最小值函数
echo "<p>";
echo max(5,1,3,7,8);                    //调用求最大值函数
?>
</body>
</html>
```

保存以上代码为“6-5.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.5 所示。



图 6.5　PHP 库函数的使用实例 1 执行结果

调用数组函数。

```
<html>
<head>
<title>PHP 库函数的使用实例 2</title>
</head>
<body>
<?
$a[0]=1;
$a[1]=3;
$a[2]=2;
$a[3]=1;
$a[4]=2;
$a[5]=1;
$a[6]=4;
$a[7]=3;
print_r(array_count_values ($a));
?>
</body>
</html>
```

将以上代码保存为“6-6.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.6 所示。

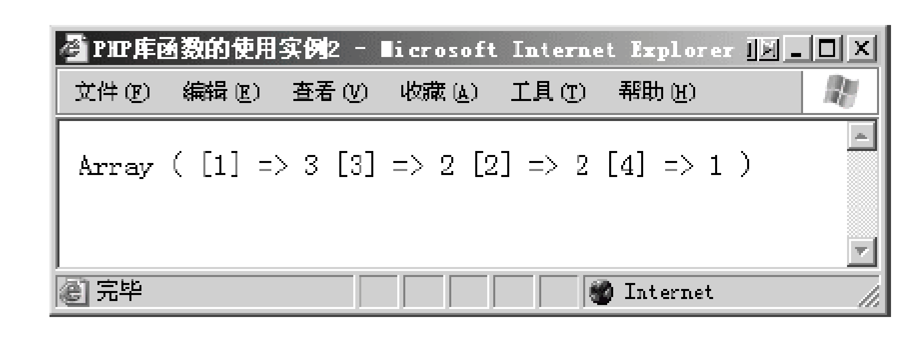


图 6.6　PHP 库函数的使用实例 2 执行结果

从上两个例子中可以看出，PHP 的库函数有的需要参数，有的不需要参数，有的还需要多个参数。所以在使用函数前，了解该函数的使用方法是很有必要的。本书将在以后章节中专门讲解相关的函数使用方法。

6.5　PHP 加载外部函数

PHP 有很多库函数，还有一些函数需要和特定的 PHP 扩展模块一起编译，否则在使用它们的时候就会得到一个致命的“未定义函数”错误。例如，要使用图像函数如 imagecreatetruecolor()，就需要在编译 PHP 的时候加上对 GD 库的支持。具体作法就是修改 php.ini 文件。找到这一行：

```
;extension=php_gd.dll
```

把这一行行首的“;”去掉，这样 PHP 解释器在启动时就会加载 GD 库函数，然后就可以像使用内部库函数一样使用 GD 库函数。

对其他外部函数的使用也是如此，要使用相应的函数，就要先加载相应的模块。有很多核心函数已包含在每个版本的 PHP 中，如字符串和变量函数等。调用 Phpinfo()函数，可以了解到 PHP 加载了哪些扩展库。同时还应该注意，很多扩展库默认就是有效的。

下面通过一个实例，来说明如何加载并使用外部函数。因为要使用 GD 库函数，所以第 1 步修改 php.ini 文件，去掉“;extension=php_gd.dll”这一行行首的“;”。另外，在这个例子中要用到 wingdings.ttf 字体，所以要把 wingdings.ttf 这个字体文件复制到 PHP 文件的同一个目录下。该实例要实现的功能是创建一个图像文件，并在图像上画出一些图形。具体代码如下：

```
<?
Header("Content-type: image/png");                       //输出一个 PNG 图片文件
$im=imagecreate(440,100);                                 //初始化图形区域
$black=imagecolorallocate($im, 0,0,0);                    //定义黑色
$white= imagecolorallocate($im, 255,255,255);             //定义白色
$yellow= imagecolorallocate($im,255,255,0);               //定义黄色
$blue = imagecolorallocate($im,0,0,255);                  //定义蓝色
$red= imagecolorallocate ($im,255,0,0);                   //定义红色
$zi= imagecolorallocate($im,255,0,255);                   //定义紫色
$color=$blue;                                             //定义$color 变量为蓝色
$font="wingdings.ttf";                                    //定义字体文件
imagefilledrectangle($im, 5, 5, 435, 95, $color);         //用蓝色画一个矩形
imagestring($im,5,7,10,"I:send",$white);                  //用白色写字符
for($i=0;$i<5;$i++)                                       //用循环画字符
    {
        imagettftext($im,40,0,90+$i*50,57,$yellow,$font,"Z");   //画出字符用黄色及字体
    }
imagestring($im,5,270,60,"to:YOU As a gift",$white);      //用白色写字符
imagestring($im,5,305,80,date(Y).".".date(m).".".date(d),$white);   //写出当前日期
imagepng($im);                                            //创建图形
imagedestroy($im);                                        //关闭图形
?>
```

保存以上代码为“6-7.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.7 所示。



图 6.7　GD 库函数使用实例执行结果

因为这一节只讲怎么使用外部函数，所以并不详细说明函数具体是怎么使用的。关于 GD 库函数

的使用，将会在第 10 章 PHP 中的图像处理中专门为读者介绍。

6.6　自定义函数

在实际进行 PHP 编程时，由于要面对的情况可能十分复杂，仅仅依靠 PHP 内置的库函数，有时往往不能实现用户所要达到的目的。这时就要用户自己构造函数来解决实际问题。PHP 允许用户使用自定义函数。那么，自定义函数应该怎么用呢？这一节就来解决如何使用自定义函数及函数的动态调用、函数的递归等问题。

6.6.1　如何自定义函数

在 PHP 中，自定义函数是一件很简单的事情。只需使用以下语法格式就可以完成对函数的自定义：

```
function functionname()
{
	statement;
	return $retval;
}
```

从以上代码中可见，要自定义函数，就是使用 function 语句，后面跟函数名加“()”。如果函数需要参数，就要把参数加在括号内。函数体部分用“{}”来包括起来，以使之和其他语句分开。大括号内是函数体，其中包括所要执行的内容、返回值等。

下面通过一个实例来说明，如何完成自定义函数的自定义。

```
<html>
<head>
<title>函数的自定义实例</title>
</head>
<body>
<?
function my_f($num1,$num2)                  //定义函数求两个数的最小公倍数
{
	if($num1>$num2)                         //如果前面数大两者互换
	{
		$temp=$num1;
		$num1=$num2;
		$num2=$temp;
	}
	$s=$num2;                               //定义变量备用
	$i=1;                                   //定义变量备用
	while($s%$num1!=0)                      //是否满足最小公倍数
	{
		$s=$num2*$i;                        //大数翻倍
		$i++;
```

```
        }
        return $s;                                  //返回结果
    }
    echo my_f(1,3);                                 //输出 1 与 3 的最小公倍数
    echo "<p>";
    echo my_f(6,8);                                 //输出 6 与 8 的最小公倍数
    echo "<p>";
    echo my_f(13,29);                               //输出 13 与 29 的最小公倍数
    echo "<p>";
    echo my_f(5,100);                               //输出 5 与 100 的最小公倍数
    echo "<p>";
    echo my_f(35,3);                                //输出 35 与 3 的最小公倍数
    echo "<p>";
    ?>
    </body>
    </html>
```

上例中，定义了一个函数，作用是通过大数翻倍法来求两个数的最小公倍数。保存以上代码为“6-8.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.8 所示。

通过上例，读者能够理解到 PHP 中是如何自定义函数的。只要掌握了方法，自定义函数其实是一件很简单的事情。

图 6.8　函数的自定义实例执行结果

6.6.2　使用自定义函数

自定义函数在完成定义后，就可以使用了。使用的方法也相当简单，就像使用 PHP 库函数一样。代码如下：

```
function functionname();
```

函数名后面加上括号，里面带上适当的参数就行了。下面继续通过一个实例来说明，在 PHP 中是如何使用自定义函数的。

```
<html>
<head>
<title>自定义函数的使用实例</title>
</head>
<body>
<?
function my_f($num1,$num2,$num3)                    //定义函数求一元二次方程的根
{
    if(($num2*$num2-4*$num1*$num3)<0)               //无实根的情况
    echo "方程没有实根！";
    elseif(($num2*$num2-4*$num1*$num3)==0)          //一个实根
    {
        echo "方程有一个实根：<p>";
        echo (-$num2+sqrt($num2*$num2-4*$num1*$num3))/(2*$num1);
```

```
        }
        else                                            //两个实根
        {
            echo "方程有两个实根：<p>";
            echo (-$num2+sqrt($num2*$num2-4*$num1*$num3))/(2*$num1);
            echo "，";
            echo (-$num2-sqrt($num2*$num2-4*$num1*$num3))/(2*$num1);
        }
    }
    echo "方程：2x<sup>2</sup>+3x+1=0 的根为：<p>";
    my_f(2,3,1);                                        //调用函数
    echo "<p>";
    echo "方程：x<sup>2</sup>+9x+1=0 的根为：<p>";
    my_f(1,-6,9);                                       //调用函数
    echo "<p>";
    echo "方程：3x<sup>2</sup>+2x+1=0 的根为：<p>";
    my_f(3,2,1);                                        //调用函数
    ?>
    </body>
    </html>
```

以上代码定义了一个函数，该函数的作用是使用公式法求一元二次方程的根。定义完成后，根据 3 种不同情况调用函数。把以上代码保存为“6-9.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.9 所示。

这一节，自定义函数的使用就介绍到这里。通过 6-8.php 和 6-9.php 两个例子，对于如何定义一个函数，如何使用自定义函数，读者应该有一个清晰的认识了。

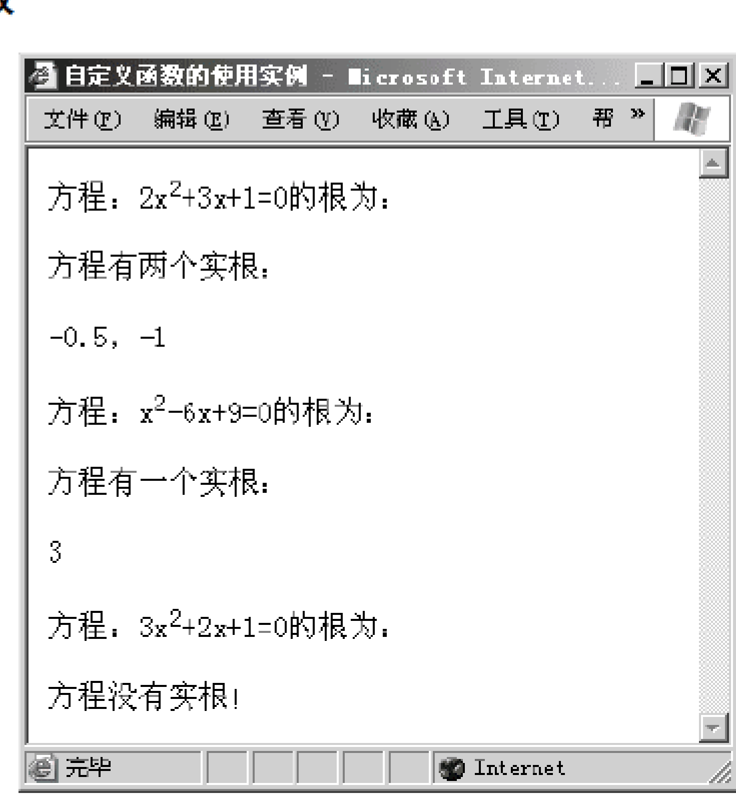


图 6.9　自定义函数的使用实例执行结果

6.6.3　函数的动态调用

由于 PHP 支持可变化的函数概念。所以如果在一个变量的名称后面加上一对圆括号“()”，那么 PHP 将去寻找与这个变量名字相同的函数。无论这个变量的数值是什么，函数都会被执行。这个过程就实现了函数的动态调用。

为了形象说明函数的动态调用这个问题，下面通过实例来介绍。

```
<html>
<head>
<title>函数的动态调用实例</title>
</head>
<body>
<?
function my_f_1($text)                              //定义函数 1
{
    echo "<font size=12pt>";                        //以 12 号字体输出文字
    echo $text;
```

```
        echo "</font>";
    }
    function my_f_2($text)                          //定义函数 2
    {
        echo "<font size=20pt>";                    //以 20 号字体输出文字
        echo "<u>";                                 //给文字加上下划线效果
        echo $text;
        echo "</u>";
        echo "</font>";
    }
    $test="my_f_1";
    $test("I LIKE PHP!");                           //动态调用 my_f_1()
    echo "<p>";
    $test="my_f_2";
    $test("用 PHP 编程，其实很简单！");              //动态调用 my_f_2()
    ?>
    </body>
    </html>
```

把以上代码保存为“6-10.php”。在 PHP 执行环境中运行以上代码，将出现如图 6.10 所示的执行结果。

以上代码先定义了两个函数，然后把函数名称赋值为变量。变量名后加上括号，PHP 就会去寻找同名的函数 my_f_1，找到后则运行，从而实现了函数的动态调用。调用 my_f_2 的过程与之类似。

通过上面的实例，相信读者对 PHP 中如何动态调用函数已经有了一定程度的了解。

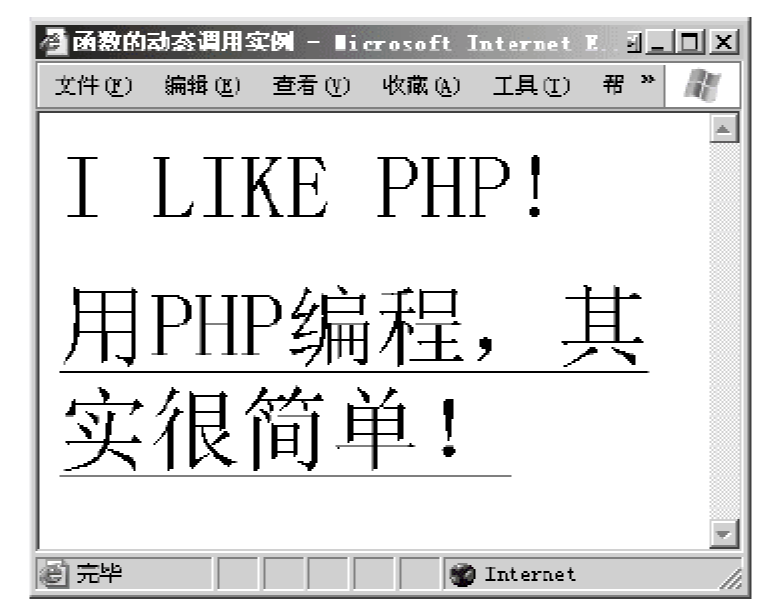


图 6.10　函数的动态调用实例执行结果

6.6.4　函数的递归

这一小节来介绍 PHP 函数的递归。那么什么是递归呢？其实递归就像读者都听过的一个歌谣那样：从前有座山，山里有座庙，庙里有个老和尚和一个小和尚。老和尚给小和尚讲故事，故事里说从前有座山，山里有座庙……就像这样无限循环。回过头来继续说递归，递归简单的说就是函数自身调用自身。

有时通过函数的递归来处理问题是十分有效的。如求斐波纳契数列第 N 项的值，如果采用传统方法效率很低。但如果使用函数的递归，解决起来就会容易的多。下面就通过例子来说明函数的递归问题。

在列出具体代码前，先来了解一下斐波纳契数列的特点。斐波纳契数列即“兔子生兔子的问题”：有一个人把一对兔子放在四面封闭的地方。假定每个月一对兔子生下另外一对。而这新的一对在二个月后就生下另外一对。这样一年后他会有多少对兔子？这里对这个数列作一改动，设第一项与第二项为 1。结果就是像这样的一组数列：1，1，2，3，5，8，13，21，34，55，89，144……即某一项为它前面两项之和。在了解了数列的特点后，下面就通过实际使用函数的递归，来解决这一问题。

```
    <html>
    <head>
    <title>函数的递归实例</title>
```

```
</head>
<body>
<?
function Fibanacci($num)                          //定义 Fibanacci 函数
{
    if($num==1 || $num==2)                        //如果为第一项和第二项
    return 1;                                     //返回值为 1
    else                                          //除 1、2 外的其他项
    {
        return Fibanacci($num-1)+Fibanacci($num-2); //递归调用前两项之和
    }
}
echo "斐波纳契数列的第 1 项为：";
echo Fibanacci(1);
echo "<p>";
echo "斐波纳契数列的第 12 项为：";
echo Fibanacci(12);
echo "<p>";
echo "斐波纳契数列的第 7 项为：";
echo Fibanacci(7);
echo "<p>";
echo "斐波纳契数列的第 20 项为：";
echo Fibanacci(20);
?>
</body>
</html>
```

保存以上代码为“6-11.php”。在 PHP 执行环境中运行以上代码，执行结果如图 6.11 所示。



图 6.11　函数的递归实例执行结果

以上代码中当参数为 1 或 2 的情况很容易理解，直接返回 1 即可。除此以外的情况直接返回了和数组定义完全相同的公式：某一项为其前两项之和，从而实现了函数的递归。

可以看出使用函数的递归解决此类问题相对于用迭代法来说是简单的、有效的。

另外，在使用函数递归时要注意以下两个问题：

- 在递归中要有使递归中止的代码，不能使递归陷入无限循环之中。
- 同时要避免递归函数调用超过 100~200 层的范围，因为可能会破坏堆栈从而使当前脚本终止。

6.7　本 章 小 结

这一章介绍了 PHP 中函数的使用方法。其中讲了什么是函数、函数的参数和返回值、PHP 库函数的使用、外部函数的加载及自定义函数。正如本章开始时提到的，函数是 PHP 的最重要的组成部分，只有熟练的使用函数，才算是真正学会了 PHP。通过本章的学习使读者对 PHP 中函数的使用有了一定了解。关于具体的函数如何使用，从第 7 章开始将用 9 章的篇幅专门为读者讲解 PHP 中各种库函数的使用。

第 7 章

PHP 的数据处理

- 怎样判断数据类型
- PHP 中常用的数学函数
- PHP 中常用的字符串函数
- PHP 中常用的数组函数

数据处理在 PHP 编程中有重要的地位，不论是编什么样的程序都少不了和各种各样的数据打交道。本章为读者介绍 PHP 中对数据的处理即 PHP 中有关数据处理的库函数。内容包括判断数据类型、常用数学函数、字符串函数、数组函数等。数据处理是编写大型 Web 应用程序的基础，只有正确地对数据进行相应的处理才能为大型应用程序提供数据基础支持。

7.1　怎样判断数据类型

在使用数据变量时，先弄清该变量属于什么类型是很有必要的。只有知道了数据变量的类型，才能对它进行相关的操作。那么 PHP 中是如何判断数据类型的呢？

在 PHP 中有专门的函数来判断数据的类型，这就是 Is 系列函数。常用的判断数据类型函数见表 7.1。

表 7.1　PHP 中的判断数据类型函数

函　数　名	作　　用	返　回　值
is_array(mixed var)	判断变量是否为数组	如果参数var是数组就返回True，否则返回False
is_bool(mixed var)	判断变量是否为布尔型	如果参数var是布尔型值即True或False就返回True，否则返回False
is_float(mixed var)	判断变量是否为浮点数	果参数var是浮点数则返回True，否则返回False
is_int(mixed var)	判断变量是否为整型变量	如果参数var为整型变量INT则返回True，否则返回False
is_null(mixed var)	判断变量是否为NULL值	如果参数var未被定义或者被设置为NULL或者虽然已经被定义但又被unset()取消定义，则返回True，否则返回False
is_numeric(mixed var)	判断变量是否为数字或者数字字符串	如果参数var为数字或者数字字符串则返回True，否则返回False
is_object(mixed var)	判断变量是否为一个对象	如果参数var为OBJECE就返回True，否则返回False
is_scalar(mixed var)	判断变量是否为一个标量（标量即最小的变量单位，只包括integer、float、string或boolean的变量，而array（数组）、object（对象）和resource（资源）则不是标量）	如果参数var为标量则返回True，否则返回False
is_string(mixed var)	判断变量是否为字符串	如果参数var为字符串返回True，否则返回False
isset(mixed var)	判断变量是否设置	如果变量存在就返回True，否则返回False。另外被设置为NULL值的变量在使用isset()时也将返回False 该函数只能用于变量，因为传递任何其它参数都将造成解析错误

表 7.1 列出了 PHP 中判断数据类型的函数。在了解了这些函数后，下面通过实例来说明这些函数是如何应用的。请参看以下代码：

```
<html>
<head>
<title>PHP 判断数据类型使用实例</title>
</head>
<body>
<?
//先设置一些变量备用
$num1=123;
$arr=array(1,2,3,4,5,6,7);
$bool1=FALSE;
$b=Null;
$num2=3.14159;
$string1="123456";
$string2="HELLO WORLD!";
class foo                                   //定义对象
{
        function do_foo()
        {
                echo "Doing foo.";
        }
}
$bar = new foo;                             //引用对象
if(is_array($arr))                          //是否为数组
echo $arr."是数组";
echo "<br>";
if(!is_array($num1))                        //是否为数组
echo $num1."不是数组！";
echo "<br>";
if(!is_bool($num1))                         //是否为布尔型
echo $num1."不是布尔型！";
echo "<br>";
if(is_bool($bool1))                         //是否为布尔型
echo $bool1."是布尔型";
echo "<br>";
if(!is_float($num1))                        //是否为浮点型
echo $num1."不是浮点型！";
echo "<br>";
if(is_float($num2))                         //是否为浮点型
echo $num2."是浮点型";
echo "<br>";
if(is_int($num1))                           //是否为整型
echo $num1."是整型";
echo "<br>";
if(!is_int($num2))                          //是否为整型
echo $num2."不是整型！";
echo "<br>";
if(is_numeric($string1))                    //是否为数字或数字字符串
echo $string1."是数字";
echo "<br>";
if(!is_numeric($string2));                  //是否为数字或数字字符串
```

```
echo $string2."不是数字！ ";
echo "<br>";
if(is_object($bar))                    //是否为对象
print_r($bar);echo "是 OBJCET";
echo "<br>";
if(!is_object($num1))                  //是否为对象
echo $num1."不是 OBJECT！ ";
echo "<br>";
if(is_scalar($num1))                   //是否为标量
echo $num1."是标量";
echo "<br>";
if(!is_scalar($arr))                   //是否为标量
echo $arr."不是标量！ ";
echo "<br>";
if(!is_string($num1))                  //是否为字符串
echo $num1."不是字符串！ ";
echo "<br>";
if(is_string($string1))                //是否为字符串
echo $string1."是字符串";
echo "<br>";
if(isset($string1))                    //是否有值
echo $string1."有值";
echo "<br>";
if(!isset($b))                         //是否有值
echo $bool1."无值或者值为 Null！ ";
echo "<br>";?>
</body>
</html>
```

保存以上代码为“7-1.php”。在 PHP 运行环境中执行以上代码，执行结果如图 7.1 所示。

通过该例子，可以了解到在 PHP 中使用相关函数，来判断数据类型是一件很简单的事情。

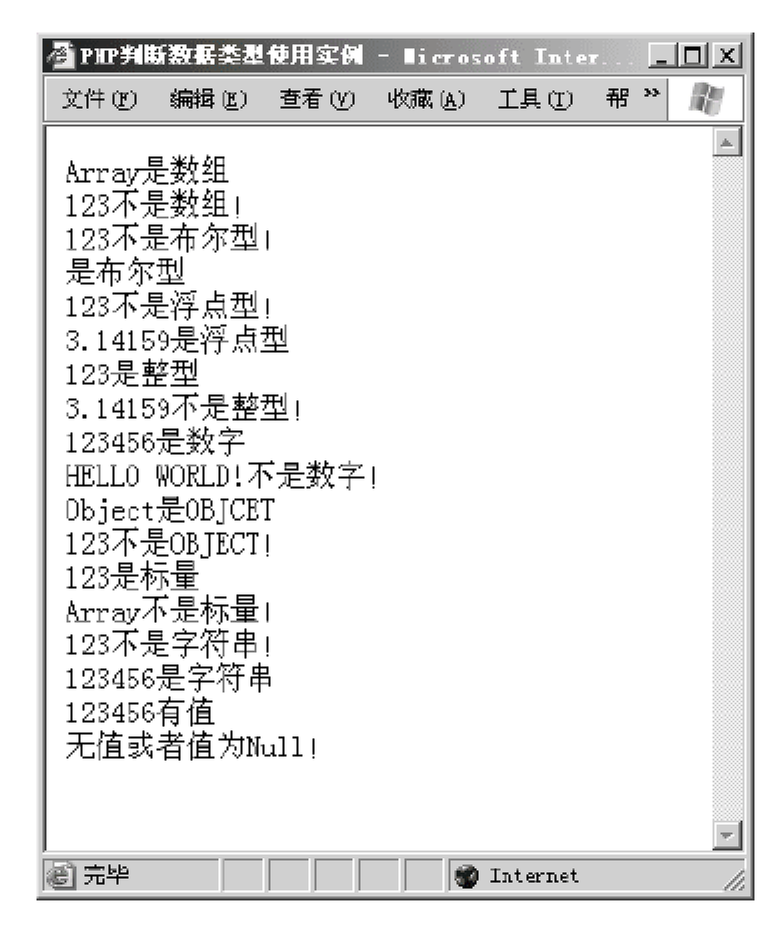


图 7.1　PHP 判断数据类型使用实例执行结果

7.2　PHP 中常用的数学函数

数学运算、数字的操作在 PHP 的数据处理中占有很大比重。无论是大型的程序，还是只有几行代码的小程序，只要牵涉到数字都需要用数学的方法进行处理。这一节就来向读者介绍 PHP 中常用的数学函数。

7.2.1　数学计算函数

这一小节为读者介绍常用的数学计算函数，详见表 7.2。

表 7.2　PHP 中常用的数学计算函数

函　数　名	作　　用	返　回　值
abs(mixed var)	绝对值函数，返回参数var的绝对值	如果参数为正，直接返回；如果为负，去掉负号后返回
ceil(float value)	进一法取整数函数，返回不小于参数value的下一个整数	如果参数为整型数，直接返回参数；如果参数为浮点型数，则返回值为参数的小数部分进一位
exp(float arg)	计算指数函数，计算e的指数。	用e作为自然对数的底2.718282返回e的arg次方值
floor(float,value)	舍去法取整数函数，返回不大于参数value的下一个整数	如果参数为整型数，直接返回参数；如果参数为浮点型数，则返回值为参数舍去小数部分后的值
log10()	计算对数函数。返回以10为底的对数	返回以10为底的对数值
sqrt(float arg)	计算平方根函数。返回参数arg的平方根	返回参数arg的平方根

在了解了这类数学计算函数后，下面通过具体的实例来向读者介绍如何使用。代码如下：

```
<html>
<head>
<title>PHP 数学计算函数使用实例</title>
</head>
<body>
<?
echo "-3 的绝对值是：";
echo abs(-3);                              //输出-3 的绝对值
echo "<P>";
echo "3 的绝对值是：";
echo abs(3);                               //输出 3 的绝对值
echo "<P>";
echo "不小于 5.5 的最小整数是：";
echo ceil(5.5);                            //输出不小 5.5 的最小整数
echo "<P>";
echo "不小于-8.2 的最小整数是：";
echo ceil(-8.2);                           //输出不小于-8.2 的最小整数
echo "<P>";
echo "e 的 3 次方是：";
echo exp(3);                               //输出 e 的 3 次方
echo "<P>";
echo "不大于 3.2 的最大整数是：";
echo floor(3.2);                           //输出不大于 3.2 的最大整数
echo "<P>";
echo "不大于-8.8 的最大整数是：";
echo floor(-8.8);                          //输出不大于-8.8 的最大整数
echo "<P>";
echo "100 以 10 为底的对数为：";
echo log10(100);                           //输出 100 以 10 为底的对数
echo "<P>";
echo "9 的平方根是：";
echo sqrt(9);                              //输出 9 的平方根
```

```
echo "<P>";
?>
</body>
</html>
```

保存以上代码为“7-2.php”。在 PHP 执行环境中运行以上代码，运行结果如图 7.2 所示。

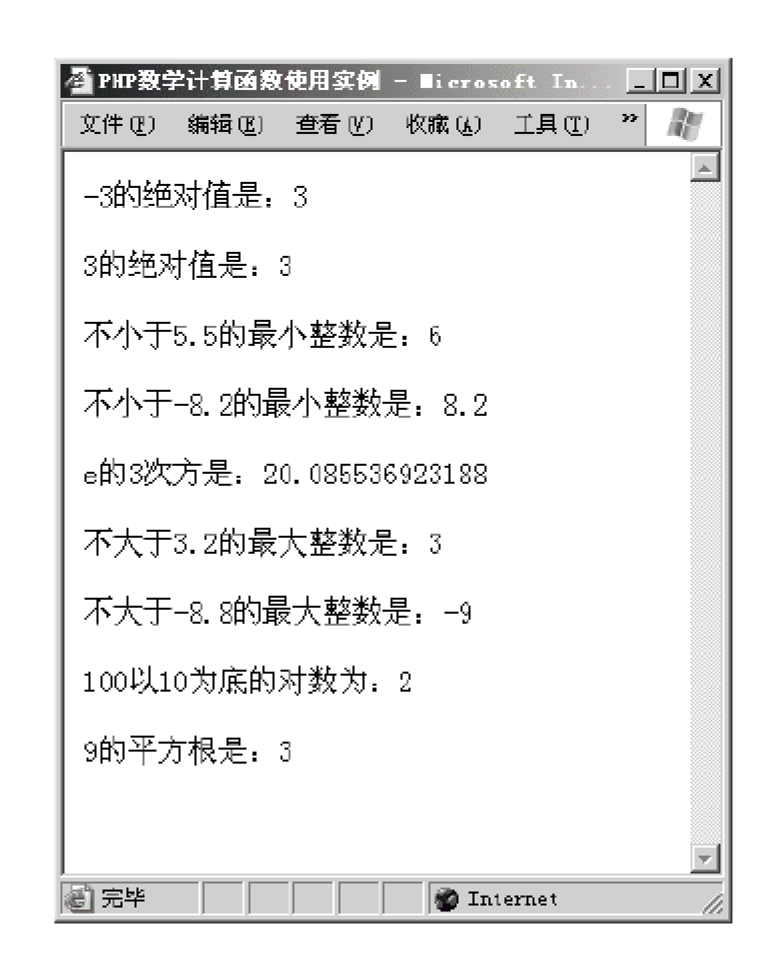


图 7.2　PHP 数学计算函数使用实例执行结果

通过上述例子及执行结果能够发现，通过 PHP 中的数学计算函数能对数据进行各种数学计算。熟练掌握这些函数将对以后的数据处理十分有益。

7.2.2　数学三角函数

这一小节来介绍常用的数学三角函数。在介绍三角函数之前先介绍一个函数：deg2rad()函数，它的作用是把角度转换为弧度。

学过三角函数的读者都知道，三角函数就是指正弦、余弦、正切、余切以及这些函数对应的反函数等。PHP 中也有类似的函数，并且表达方法几乎与数学的表达方法一样。PHP 中的常用三角函数有以下几个，见表 7.3。

表 7.3　PHP 中常用的三角函数

函　数　名	作　　用	返　回　值
sin(float arg)	正弦函数	返回参数arg的正弦值，参数arg为弧度
cos(float arg)	余弦函数	返回参数arg的余弦值，参数arg为弧度
tan(float arg)	正切函数	返回参数arg的正切值，参数arg为弧度
asin(float arg)	反正弦函数	返回参数arg的反正弦值，参数arg为弧度。函数asin()是函数sin()的反函数
acos(float arg)	反余弦函数	返回参数arg的反余弦值，参数arg为弧度。函数acos()是函数cos()的反函数
atan(float arg)	反正切函数	返回参数arg的反正切值，参数arg为弧度。函数atan()是函数tan()的反函数

下面通过两个实例来说明在 PHP 中如何使用这些三角函数。

首先来看看简单三角函数的使用。代码如下：

```
<html>
<head>
<title>PHP 三角函数使用实例 1</title>
</head>
<body>
<?
echo "30 度角的正弦值为：";
echo sin(pi()/6);                                   //计算正弦
echo "<p>";
echo "270 度角的正弦值为：";
echo sin(pi()/2*3);
```

```
echo "<p>";
echo "60 度角的余弦值为：";
echo cos(pi()/3);                          //计算余弦
echo "<p>";
echo "0 度角的余弦值为：";
echo cos(0);
echo "<p>";
echo "45 度角的正切值为：";
echo tan(pi()/4);                          //计算正切
echo "<p>";
echo "90 度角的正切值为：";
echo tan(pi()/2);
echo "<p>";
echo "30 度角的反正弦值为：";
echo asin(pi()/6);                         //计算反正弦
echo "<p>";
echo "60 度角的反余弦值为：";
echo acos(pi()/3);                         //计算反余弦
echo "<p>";
echo "45 度角的反正切值为：";              //计算反正切
echo atan(pi()/4);
echo "<p>";?>
</body>
</html>
```

保存以上代码为“7-3.php”。在 PHP 执行环境中运行以上代码，执行结果如图 7.3 所示。

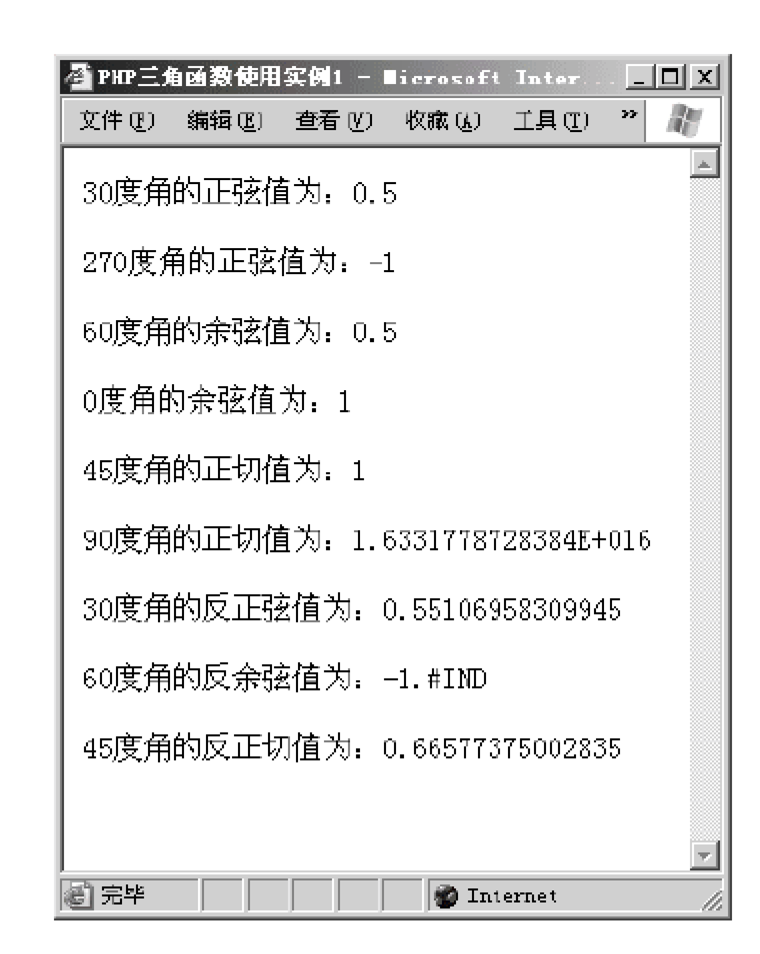


图 7.3　PHP 三角函数使用实例 1 执行结果

接下来，使用三角函数结合 GD 函数来实现画图功能（更多关于 GD 库函数的使用方法请参看本书第 10 章）。

```
<?
Header("Content-type: image/png");                  //输出文件头为 PNG 图片
$im = imagecreate(400,400);                         //使用 GD 库函数创建区域
$black = Imagecolorallocate($im, 0,0,0);            //使用 GD 库函数定义黑色
$white = Imagecolorallocate($im, 255,255,255);      //使用 GD 库函数定义白色
$yellow = Imagecolorallocate($im,255,255,0);        //使用 GD 库函数定义黄色
$blue = Imagecolorallocate($im,0,0,255);            //使用 GD 库函数定义蓝色
$red = Imagecolorallocate($im,255,0,0);             //使用 GD 库函数定义红色
imagefilledrectangle($im, 5, 5, 395, 395, $yellow); //使用 GD 库函数画矩形
for($i=1;$i<360;$i++)                               //通过循环画点
{
     $temp=150*sin(2*(pi()/180)*$i);                //通过三角函数计算值
     $x=$temp*cos((pi()/180)*$i)+200;               //通过三角函数计算点的横坐标
     $y=$temp*sin((pi()/180)*$i)+200;               //通过三角函数计算点的纵坐标
     imagesetpixel ($im,$x,$y,$red);                //通过 GD 库函数画点
     $temp=150*cos(2*(pi()/180)*$i);                //通过三角函数计算第 2 个值
     $x=$temp*cos((pi()/180)*$i)+200;               //通过三角函数计算点的横坐标
     $y=$temp*sin((pi()/180)*$i)+200;               //通过三角函数计算点的纵坐标
     imagesetpixel ($im,$x,$y,$blue);               //通过 GD 库函数画点
```

```
}
ImagePNG($im);                                    //输出 PNG 图片
ImageDestroy($im);                                //清空图片
?>
```

在运行这个代码以前，要加载 GD 库函数到 PHP 解释器中。保存以上代码为“7-4.php”。然后在 PHP 执行环境中运行以上代码，执行结果如图 7.4 所示。

代码运行之后，在屏障上画出漂亮的花的图案。这一方面说明了 PHP 功能的强大，同时也是 PHP 三角函数使用带来的奇妙效果。不过请不要过分关注其中 GD 库函数的使用方法，关于其中的 GD 库函数，将在第 10 章 PHP 中的图像处理中详细为读者做介绍。

PHP 的三角函数使用，这一节就介绍到这里。

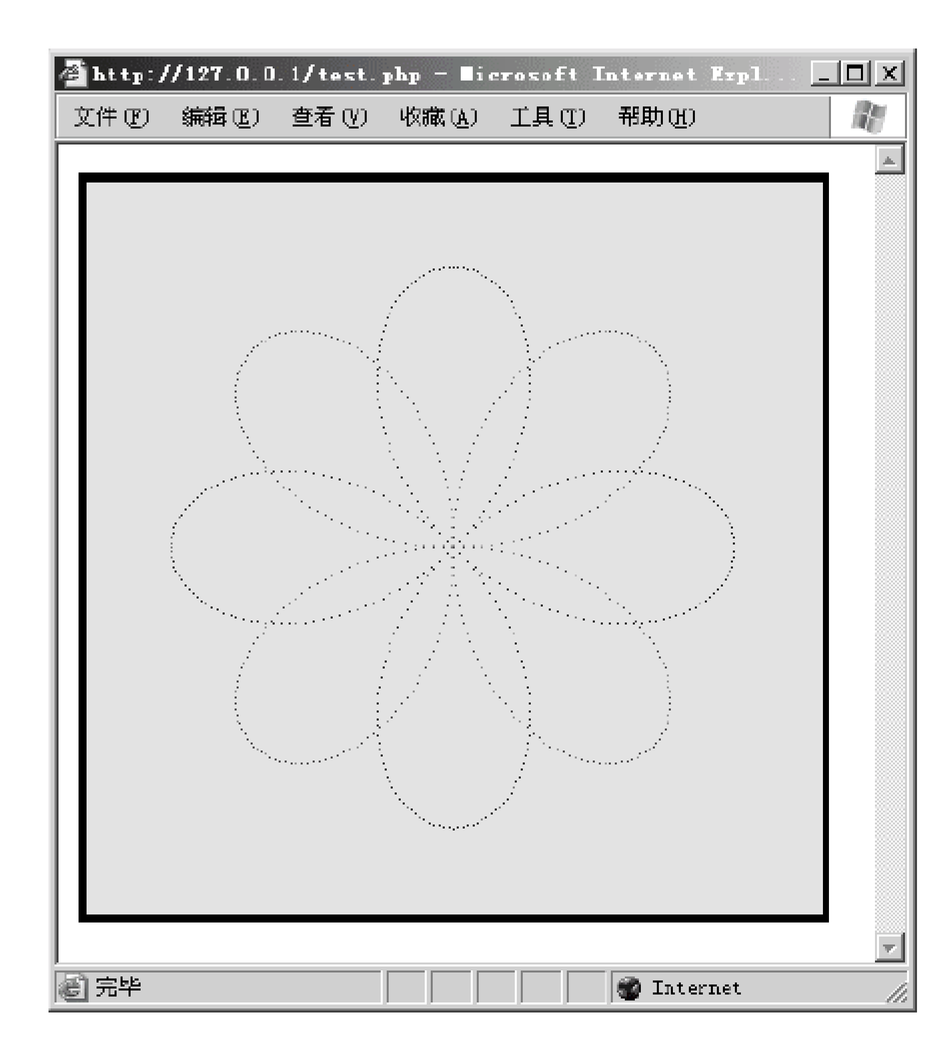


图 7.4　PHP 三角函数使用实例 2 执行结果

7.2.3　很有用的最值函数

如果给出一组数据让求最大（小）值，应该怎么做？在通常的编程语言中，要先把一组数赋值给一个数组，然后再对数组进行冒泡法排序。最后再把得出的结果赋值给一个变量。真是相当麻烦。在 PHP 中这一切就变得很简单了，因为 PHP 库函数中就为用户提供了求一组数据最值的函数。PHP 中的最值函数有以下几种：

- Max(Number Arg1, Number Arg2)，求最大值函数，返回参数中数值最大的值。如果仅有一个参数且为数组，Max()返回该数组中最大的值。如果第 1 个参数是整数、字符串或浮点数，则至少需要两个参数而 Max()会返回这些值中最大的一个。可以比较无限多个值。
- Min(Number Arg1, Number Arg2)，求最小值函数，返回参数中数值最小的值。如果仅有一个参数且为数组，Min()返回该数组中最小的值。如果第 1 个参数是整数、字符串或浮点数，则至少需要两个参数而 Min()会返回这些值中最小的一个。可以比较无限多个值。

下面通过两个实例来介绍在实际的编程中如何使用这两个函数及注意事项。

1．Max()函数使用实例

```
<html>
<head>
<title>max()函数使用实例</title>
</head>
<body>
<?
echo "1,3,5,6,7 中数值最大的是：";
echo max(1,3,5,6,7);                       //比较多个数值
echo "<p>";
echo "数组 array(2,4,5)中最大的值是：";
echo max(array(2, 4, 5));                  //比较一个数组
echo "<p>";
```

```
echo "0 与“hello”中最大的是：";
echo max(0, 'hello');                              //比较数及字符串
echo "<p>";
echo "“hello”与 0 中最大的是：";
echo max('hello', 0);
echo "<p>";
echo "-1 与“hello”中最大的是：";
echo max(-1, 'hello');
echo "<p>";
//对多个数组，max 从左向右比较。
//因此在本例中：2 == 2，但 4 < 5
echo "数组 array(2,4,8)与数组 array(2,5,7)中最大的是：";
$val=max(array(2,4,8), array(2,5,7));
for($i=0;$i<count($val);$i++)
{
        echo $val[$i];
        echo "，";
}
echo "<p>";
//如果同时给出数组和非数组作为参数，则总是将数组视为
//最大值返回
echo "数组 array(2,5,7)与“string”和 42 中最大的是：";
$val = max('string', array(2, 5, 7), 42);
for($i=0;$i<count($val);$i++)
{
        echo $val[$i];
        echo "，";
}
?>
<body>
</html>
```

保存以上代码为“7-5.php”。在 PHP 执行环境中运行以上代码，执行结果如图 7.5 所示。



图 7.5　max()函数使用实例执行结果

从以上实例及结果中可以看到 Max()函数在不同的情况下是如何运行的。

2．Min()函数使用实例

```
<html>
<head>
<title>min()函数使用实例</title>
</head>
<body>
<?
echo "1,3,5,6,7 中数值最小的是：";
echo min(1,3,5,6,7);                               //比较多个数值
echo "<p>";
echo "数组 array(2,4,5)中最小的值是：";
echo min(array(2, 4, 5));                          //比较一个数组
```

```
echo "<p>";
echo "0 与“hello”中最小的是：";
echo min(0, 'hello');                                   //比较数及字符串
echo "<p>";
echo "“hello”与 0 中最小的是：";
echo min('hello', 0);
echo "<p>";
echo "-1 与“hello”中最小的是：";
echo min(-1, 'hello');
echo "<p>";
//对多个数组，min 从左向右比较。
//因此在本例中：2 == 2，但 4 < 5
echo "数组 array(2,4,8)与数组 array(2,5,7)中最小的是：";
$val=min(array(2,4,8), array(2,5,7));
for($i=0;$i<count($val);$i++)
{
    echo $val[$i];
    echo "，";
}
echo "<p>";
//如果同时给出数组和非数组作为参数，则不可能返回数组
//因为数组被看作是最大的。最大值返回
echo "数组 array(2,5,7)与“string”和 42 中最小的是：";
$val = min('string', array(2, 5, 7), 42);
echo $val;
?>
<body>
</html>
```

保存以上代码为“7-6.php”。在 PHP 执行环境中运行以上代码，就会看到如图 7.6 所示的执行结果。

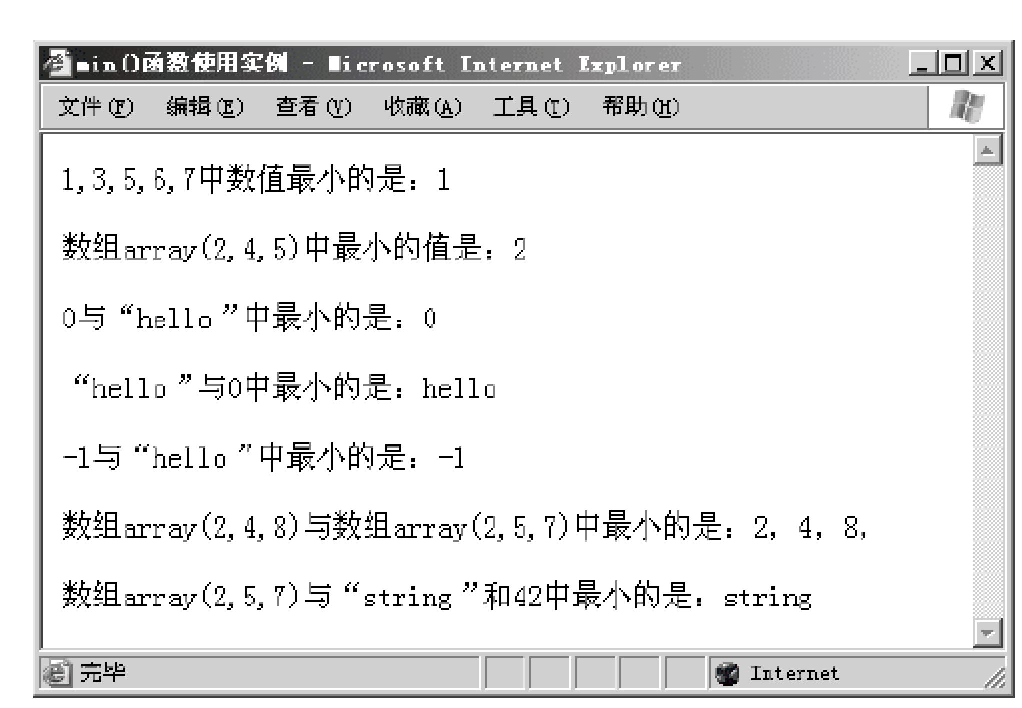


图 7.6　min()函数使用实例执行结果

通过以上实例及实例执行结果，可以了解到 Min()函数的运行机理，及在不同的情况下的返回值情况。在使用这两个函数时要注意以下问题：

- 两个函数都是从左向右比较。所以，如果相同的两个值比较优先返回左边的值（见上面两个例子中的 HELLO 字符串与 0 比较的情况）。
- 如果同时给出数组和非数组作为参数，则数组被认为是最大的。所以 Max()函数必定返回数组；而 Min()函数必定不返回数组。

这一小节中 PHP 的最值函数就为读者介绍到这里。

7.2.4　产生随机数函数

在进行 PHP 编程的时候，有时需要一些随机的数字。比如用户身份验证，为了防止站外提交，生成一组随机数字，然后在后台判断。如果随机数与系统要求的不一样就不允许其他操作。那么应该如

何生成随机数呢？PHP 中有专门的随机数函数。

Rand([int min, int max])函数，产生一个随机数。随机数的范围在参数 Min 与 Max 之间。如果省略参数，则返回 0 到 RAND_MAX 之间的伪随机整数。下面通过一个实例说明 Rand()函数是如何使用的。

```
<html>
<head>
<title>rand()随机函数使用实例</title>
</head>
<body>
<?
echo "生成一位随机数：";
echo rand(1,9);                              //生成一位随机数
echo "<p>";
echo "生成无参数随机数：";
echo rand();                                 //无参数随机数 1
echo "<p>";
echo "生成无参数随机数 2：";
echo rand();                                 //无参数随机数 2
echo "<p>";
echo "生成四位随机数：";
echo rand(1000,9999);                        //生成一个四位随机数
?>
</body>
</html>
```

保存以上代码为“7-7.php”。在 PHP 运行环境中执行以上代码，执行结果格式如图 7.7 所示（因生成数为随机数，所以结果数值应该不一定相同）。

通过实例可见，在 PHP 中生成一个随机数是一件很简单的事情。

在使用 rand()函数时有一点需要注意。在 PHP 4.2 以前的版本中，要想使用 rand()函数必须先使用 srand()函数为 rand()生成种子，然后才能使用 rand()函数。不过新版本的 PHP 已经取消了这种设定。可以直接使用 rand()随机函数。

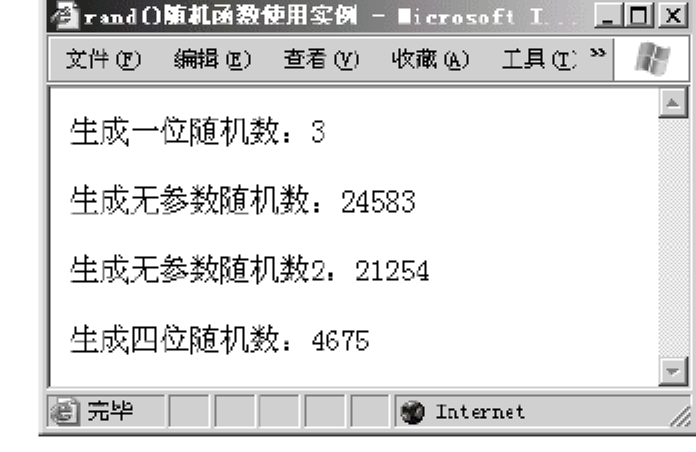


图 7.7　rand()随机函数使用实例执行结果

7.2.5　进制转换函数

在实际工作中，有时需要进行进制的转换。比如把十进制转换为二进制，或者把十六进制的数转换为十进制等。PHP 的库函数中也为用户准备好了这样的函数。PHP 中的进制转换函数一共有以下 6 种，见表 7.4。

表 7.4　PHP 中的进制转换函数

函　数　名	作　　用	返　回　值
bindec(string binary_string)	二进制转换为十进制	返回参数binary_string二进制数对应的十进制等价值
decbin(int number)	十进制转化为二进制	返回参数int十进制数对应的二进制等价值

续表

函　数　名	作　　用	返　回　值
dechex(int number)	十进制转化为十六进制	返回参数int十进制数对应的十六进制等价值
decoct(int number)	十进制转化为八进制	返回参数int十进制数对应的八进制等价值
hexdec(string hex_string)	十六进制转化为十进制	返回参数hex_string十六进制数对应的十进制等价值
ctdec(string octal_string)函数	八进制转换为十进制	返回参数octal_string八进制数对应的十进制等价值

通过以上介绍可以发现，十进制可以直接和二进制、八进制、十六进制进行转换。同时二进制、八进制、十六进制也可以直接转换为十进制。但是它们之间的转换，就只有通过十进制数来进行了。

下面通过一个实例来说明，在 PHP 编程中是如何使用这些进制转换函数的。具体过程参看如下代码：

```
<html>
<head>
<title>进制转换函数使用实例</title>
</head>
<body>
<?
//定义变量备用
$bin="11001";
$dec=100;
$hex="12F";
$oct="77";
echo "二进制数".$bin."对应的十进制数为：";
echo bindec($bin);                          //二进制转换为十进制
echo "<p>";
echo "十进制数".$dec."对应的二进制数为：";
echo decbin($dec);                          //十进制转换为二进制
echo "<p>";
echo "十进制数".$dec."对应的八进制数为：";
echo decoct($dec);                          //十进制转换为八进制
echo "<p>";
echo "十进制数".$dec."对应的十六进制数为：";
echo dechex($dec);                          //十进制转换为十六进制
echo "<p>";
echo "八进制数".$oct."对应的十进制数为：";
echo octdec($oct);                          //八进制转换为十进制
echo "<p>";
echo "十六进制数".$hex."对应的十进制数为：";
echo hexdec($hex);                          //十六进制转换为十进制
echo "<p>";
echo "二进制数".$bin."对应的八进制数为：";
echo decoct(bindec($bin));                  //二进制转换为八进制
echo "<p>";
echo "二进制数".$bin."对应的十六进制数为：";
echo dechex(bindec($bin));                  //二进制转换为十六进制
echo "<p>";
echo "八进制数".$oct."对应的二进制数为：";
```

```
echo decbin(octdec($oct));                    //八进制转换为二进制
echo "<p>";
echo "八进制数".$oct."对应的十六进制数为：";
echo dechex(octdec($oct));                    //八进制转换为二进制
echo "<p>";
echo "十六进制数".$hex."对应的二进制数为：";
echo decbin(hexdec($hex));                    //十六进制转换为二进制
echo "<p>";
echo "十六进制数".$hex."对应的二八制数为：";
echo decoct(hexdec($hex));                    //十六进制转换为八进制
echo "<p>";
?>
</body>
</html>
```

保存以上代码为“7-8.php”。在 PHP 执行环境中运行以上代码，执行结果如图 7.8 所示。

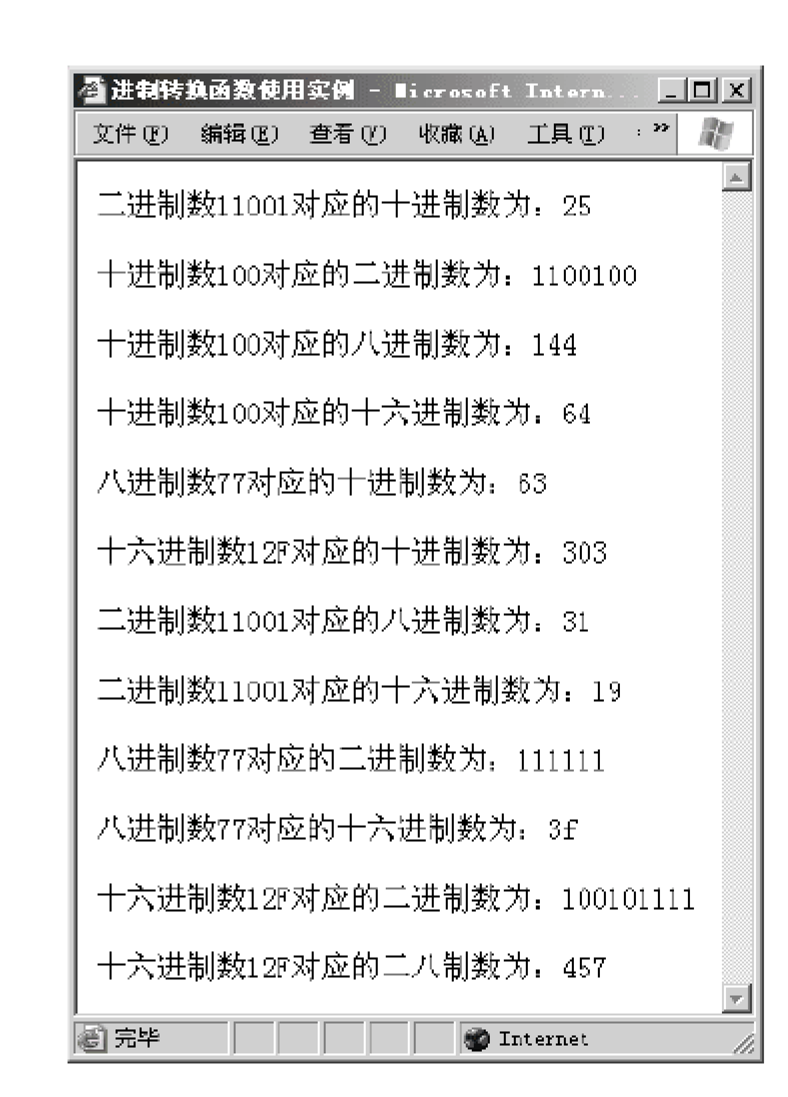


图 7.8　进制转换函数使用实例执行结果

通过以上实例及执行结果可以发现。虽然 PHP 中不能实现二进制、八进制、十六进制之间的直接转换。但是通过十进制作为中间变量，还是能方便地实现这样的目的的。

除了这 6 个函数以外，PHP 中还有一个函数：base_convert(string number,Int frombase ,int to base)可以实现任意进制之间的转换。它的返回值为一字符串，包含 Number 以 Tobase 进制的表示。Number 本身的进制由 frombase 指定。frombase 和 tobase 都只能在 2~36 之间（包括 2 和 36）。高于十进制的数字用字母 a~z 表示。如 a 表示 10，b 表示 11 以及 z 表示 35。

下面通过一个实例来说明如何使用 Base_convert()函数来进行进制转换。

```
<html>
<head>
<title>base_convert()函数使用实例</title>
</head>
<body>
<?
//定义变量备用
$bin="11001";
$hex="12F";
$oct="77";
echo "二进制数".$bin."转为十六进制数为：";
echo base_convert($bin,2,16);
echo "<p>";
echo "八进制数".$oct."转为二进制数为：";
echo base_convert($bin,8,2);
echo "<p>";
```

```
echo "十六进制数".$hex."转为八进制数为：";
echo base_convert($bin,16,8);
?>
</body>
</html>
```

保存以上代码为“7-9.php”。在 PHP 执行环境中运行以上代码，其执行结果如图 7.9 所示。

从执行结果中可以发现，使用 base_convert()函数对数据的进制进行转换，与使用进制转换系列函数转换的结果一模一样。说明函数正常运行。

这一小节 PHP 中的进制转换函数就介绍到这里。

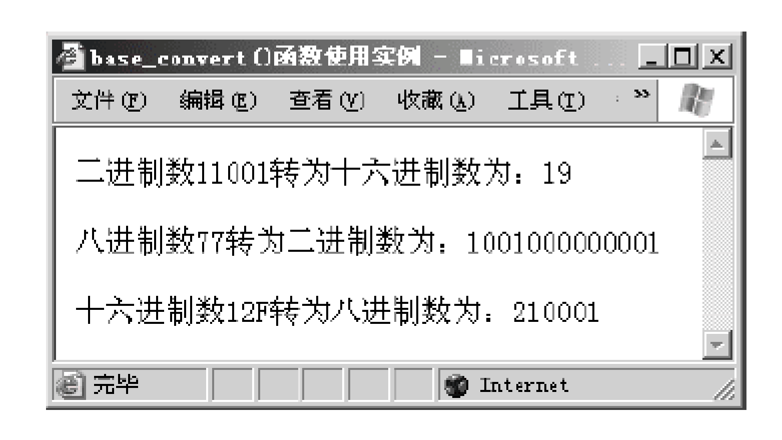


图 7.9　base_convert()函数使用实例执行结果

7.2.6　其他数学函数

除了以上 5 小节介绍的数学函数之外，PHP 中还有以下几个常用的数学函数：hypot()函数、pi()函数等。下面分别为读者介绍。

hypot(float x, float y)函数，计算一直角三角形斜边的长度。返回值为 sqrt(x*x+y*y)。

pi()函数，此函数无参数，返回圆周率的近似值：3.1415926535898。当然也可以使用系统预定义常量：M_PI 来代替这个函数。

这两个函数的使用方法相对单一，下面分别通过例子来介绍如何使用这两个函数。

1．hypot()函数使用实例

```
<html>
<head>
<title>hypot()函数使用实例</title>
</head>
<body>
<?
$a=3;                                              //定义变量
$b=4;                                              //定义变理
echo "直角三角形的一直角边为：".$a;
echo "<br>";
echo "另一直角边为".$b."则此三角形的斜边为：";
echo hypot($a,$b);                                 //计算三角形斜边
echo "<p>";
$a=5;                                              //定义变量
$b=12;                                             //定义
echo "直角三角形的一直角边为：".$a;
echo "<br>";
echo "另一直角边为".$b."则此三角形的斜边为：";
echo hypot($a,$b);
?>
</body>
</html>
```

保存以上代码为“7-10.php”。在 PHP 执行环境中运行以上代码，其执行结果如图 7.10 所示：

从上面的例子及执行结果可见 hypot()函数，实际上是返回了 sqrt($a*$a+$b*$b)。根据数学中的勾股定理，实际上就是直角三角形的斜边。

图 7.10　hypot()函数使用实例执行结果

2．pi()函数使用实例

```
<html>
<head>
<title>pi()函数使用实例</title>
</head>
<body>
<?
echo "圆周率的值为：";
echo pi();                                          //利用 pi()函数输出圆周率
echo "<p>";
$r=10;
echo "有一个圆的半径为".$r."Cm，则它的周长为：";
echo number_format(2*$r*pi(),2)."Cm";               //计算圆的周长并格式化输出
echo "<br>";
echo "此圆的面积为:";
echo number_format(pi()*$r*$r,2)."平方 Cm";          //计算圆的面积并格式化输出
?>
</body>
</html>
```

保存以上代码为“7-11.php”。在 PHP 执行环境中运行该文件，执行结果如图 7.11 所示。

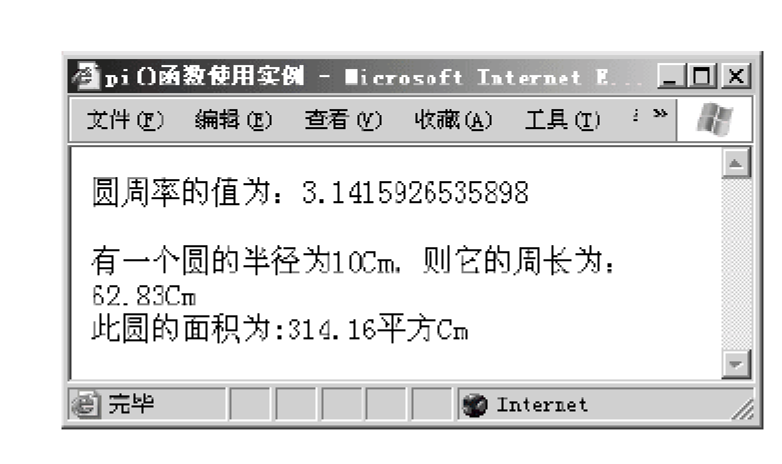


图 7.11　pi()函数使用实例执行结果

从以上例子中可以了解到 pi()函数的运行机理及返回值情况。

至此 PHP 中的数学函数就基本上介绍完了，7.3 节将会为读者介绍 PHP 中字符串处理函数。

7.3　PHP 中常用的字符串函数

字符串操作的重要性一点也不亚于数学计算。在 PHP 编程过程中，不管是进行文本的处理还是字符的操作都离不开字符串。大量的信息都是用字符串来存储的。所以说只有善于处理字符串才能称得上是一个合格的程序员。这一节来讲解 PHP 中常用的字符串函数。

7.3.1　取得字符串长度

要想处理一个字符串，第 1 步就要了解该字串的长度。在其他一些编程语言（如 Javascript）中，

字符串的长度通常都是作为字符串的一个属性出现。而 PHP 中则需要通过相应的函数来取得，这个函数就是 strlen(string)。

strlen(string)函数，取得字符串的长度。参数 string 为字符串变量，返回值为表示字符串的整型变量。空字符串将返回“0”。

下面通过一个实例来说明 strlen()函数是如何起作用的。

```
<html>
<head>
<title>取得字符串的长度</title>
</head>
<body>
<?
$s="asdfghjkl";                          //定义字符串
$s2="I love this game!";                 //定义字符串
echo "字符串".$s."的长度为：";
echo strlen($s);                         //取得字符串的长度
echo "<P>";
echo "字符串".$s2."的长度为：";
echo strlen($s2);                        //取得字符串的长度
?>
</body>
</html>
```

保存以上代码为“7-12.php”。在 PHP 执行环境中运行以上代码，将会看到如图 7.12 所示的执行结果。

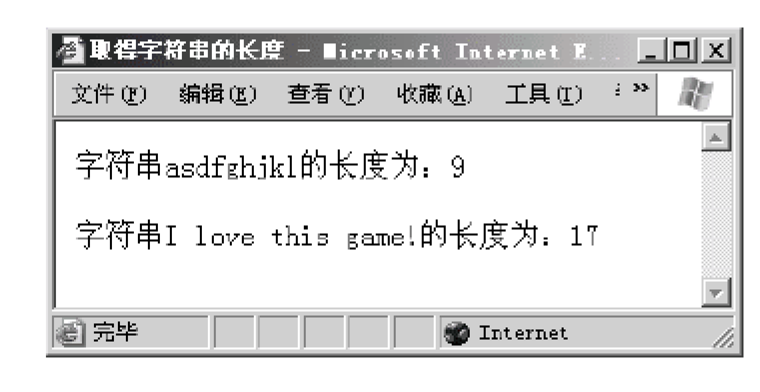


图 7.12　取得字符串的长度实例执行结果

在使用 strlen()函数时必须注意一个问题，当参数为西文字符时，能正确返回字符串的长度。而当参数为中文时，就得不到预想的结果。一个中文字符将被当作两个西文字符来处理。所以在使用 strlen()函数时要注意：参数只能是西文字符而不能是中文字符或者含有中文字符。

7.3.2　输出字符串

PHP 中用于输出字符串的函数有以下几种：

echo 函数，用于字符串的输出。echo 在前面的例子中已经多次用到过。从严格意义上来讲 echo 并不是一个真正的函数，使用的时候，不需要加“()”。另外也没有返回值。

print(string)函数，输出字符串。参数 string 为字符串变量或者常量。函数返回一个布尔值。当执行成功时返回 True，反之返回 False。

printf(string format,mixed[args])函数，将字符串格式化输出。输出格式依照参数 format 的内容，具体内容见下面的 sprintf()格式化字符串函数。

sprintf(string format,mixed[args])函数，格式化字符串。参数 format 是转换的格式，以百分比符号%开始到转换字符为止。

格式的内容按照顺序如下。

- 填空字元：0 表示空格填 0；空格是内定值。
- 对齐方式：内定值为向右对齐，负号表示向左对齐。
- 栏位宽度：为最小宽度。
- 精确度：指在小数点后的浮点数位数。
- 型态：如表 7.5 所示有以下几种。

表 7.5　格式化字符串的转换字符型态

%	印出百分比符号不转换
b	整数转成二进制
c	整数转成对应的ASCII字符
d	整数转成十进制
f	倍精确度数字转成浮点数
o	整数转成八进制
s	整数转成字串
x	整数转成小写十六进制
X	整数转成大写十六进制

因为 echo 与 print()函数相对简单，这里就不多做介绍，重点来说一下 sprintf()这个格式化函数。关于如何使用 sprintf()函数，先来看一个经典的实例：

```
<?
$money1=68.75;
$money2=54.35;
$money=$money1+$money2;                //此时变量$money 值为"123.1"
$formatted=sprintf("%01.2f",$money);   //此时变量$formatted 值为"123.10"
?>
```

上述代码中的“%01.2f”是什么意思呢？

首先这个“%”符号是开始的意思，它写在最前面表示指定格式要开始了。也就是“起始字符”，直到出现“转换字符”为格式终止。

然后跟在%符号后面的是“0”。这个零是“填空字元”，表示如果位置空着就用“0”来填满。

在 0 后面的是“1”，这个“1”是规定：小数点前面的数字占位要有 1 位以上。如果把“1”改成“2”，则$formatted 的值将为“01.23”。因为，在小数点前面的数字只占了 1 位，按照上面所规定的格式，小数点前数字应该占 2 位，现在只有 1 位，所以用“0”来填满。

介绍到这里，在%01 后面的“.2”（点 2）就很好理解了，它的意思是规定：小数点后的数字必须占 2 位。如果这时候$money 的值为“1.234”，则$formatted 的值将为“1.23”。之所以去掉了最后的“4”，是因为按照上面的规定在小数点后面必须且仅能占 2 位。可是$money 的值中，小数点占了 3 位，所以最后的“4”被去掉了只剩下“23”。

最后，以“f”这个“转换字符”结尾。

其他转换字符请自行参考表 7.5 所列出的转换字符。

关于对齐，如果在%起始符号后面加上“-”（负号）则会把数字以向右对齐的方式进行处理。

下面通过一个例子来说明 sprintf()格式化函数的使用方法：

```
<html>
<head>
<title>sprintf()函数使用实例</title>
</head>
<body>
<?
$s="123.321";                                   //定义字符串
echo $s."的原始值：".$s;
echo "<p>";
$temp=sprintf("%-1.1f",$s);                     //进行格式化
echo $s."经过格式化后的值：".$temp;
echo "<p>";
$s="12 3 5";                                    //定义字符串
echo $s."的原始值：".$s;
echo "<p>";
$temp=sprintf("%0-b",$s);                       //进行格式化
echo $s."经过格式化后的值".$temp;
?>
</body>
</html>
```

保存以上代码为“7-13.php”。在 PHP 执行环境中运行以上代码，执行结果如图 7.13 所示。

具体分析以上代码。

$s 的原始值为 123.321，“%-1.1f”表示左对齐，小数点前后各有一位数字。但小数点前已经有三位了所以保留三位。小数点后面的三位则去掉后两位保留一位。

第 2 个$s 的原始值为“12 3 5”，先去掉空格，然后左对齐，再把得到的十进制整数转化为二进制。空格前只有一个 12，所以对 12 进行十进制到二进制的转化，结果为 1100，即 8（2 的 3 次方）+4（2 的二次方）。

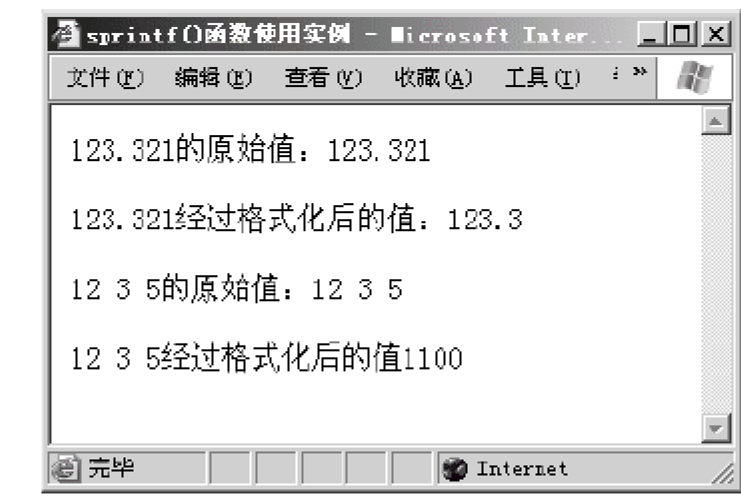


图 7.13　sprintf()函数使用实例执行结果

7.3.3　截取字符串

在进行字符串处理的时候，有时需要对字符串进行截取。在 PHP 中要做到这一点也是很简单的，因为有一个专门的函数供用户调用，这个函数就是 substr()。

substr(string string,int start,int [length])函数，截取字符串。参数 string 为字符串变量，参数 start、length 为整型变量。函数返回字符串 string 从 start 开始的 length 个字符。如果 length 为空，则从 start 截取到字符串结束。如果 start 为负数，则从字符串末尾开始截取。如果 length 为负则表示截取到倒数第 length 个字符。

下面通过一个实例来说明，如何使用 substr()函数来截取字符串。

```
<html>
<head>
<title>截取字符串</title>
</head>
<body>
<?
$s="asdfghjkl";                              //定义字符串
$s2="I love this game!";                     //定义字符串
echo "字符串".$s;
echo "<br>";
echo "从 3 开始的 5 个字符为：";
echo substr($s,3,5);                         //截取字符串
echo "<P>";
echo "字符串".$s2;
echo "<br>";
echo "从 2 开始的 0 个字符为：";
echo substr($s2,2);                          //截取字符串
echo "<p>";
echo "字符串".$s2;
echo "<br>";
echo "从-5 开始的字符为：";
echo substr($s2,-5);                         //截取字符串
echo "<p>";
echo "字符串".$s2;
echo "<br>";
echo "从 2 开始的-6 个字符为：";
echo substr($s2,2,-6);                       //截取字符串
?>
</body>
</html>
```

保存以上代码为“7-14.php”。在 PHP 执行环境中运行以上代码，其执行结果将如图 7.14 所示。

上面代码中的前两次截取都很好理解，不多解释。重点来解释一下后面两次截取。

substr($s2,-5)，意思是从尾端截取 5 个到末尾，所以返回“game!”。

substr($s2,2,-6)，意思是从第 2 个开始截取到倒数第 6 个，所以返回“love this”。



图 7.14 截取字符串实例执行结果

7.3.4 按特定字符切开字符串

在进行字符串操作过程中，有时需要把一个较长的字符串按照特定的字符分割成若干个短的字符。如文件本型数据库，通常把一行当作一条记录。而一条记录中则通过特定的标记字符来分别存放不同的字段。对这样的记录，操作时就要按这些特定的标记字符把一行记录分割开。PHP 中提供了这样的

函数：

- ❑ explode(string separator,string string[,int limit])，切开字符串函数。本函数将字符串依指定的字符或字符串 separator 分开，如果使用了 limit 参数则返回的数组包含最多 limit 个元素，而最后那个元素将包含 string 的剩余部分。函数的返回值是以返回字符串为元素的字符串数组。
- ❑ split(string pattern,string stirng[,int limit])，用正则表达式把字符串分割到数组中。本函数返回一个字符串数组，每个元素为 string 经区分大小写的正则表达式 pattern 作为边界分割出的子串。如果设定了 limit，则返回的数组最多包含 limit 个元素，而其中最后一个单元包含了 string 中剩余的所有部分。如果出错，则 split()返回 False。
- ❑ strtok(string str,string token)，切开字符串函数。本函数将传回字符串 str 依据 token 的值切成一小段一小段的字符串。

下面通过一个实例来介绍这些分割字符串函数是如何使用的。

```
<html>
<head>
<title>切开字符串使用实例</title>
</head>
<body>
<?
$s="123|456|789";                          //定义字符串
$s2="|";                                   //定义子字符串
echo "字符串".$s;
echo "<br>";
echo "使用 explode 方法分割开：";
$temp=explode($s2,$s);                     //使用 explode 截取字符串
for($i=0;$i<count($temp);$i++)             //循环显示返回的数组元素
echo $temp[$i]."，";
echo "<P>";
echo "字符串".$s;
echo "<br>";
echo "使用 explode 方法加上参数 2 分割开：";
$temp=explode($s2,$s,2);                   //使用 explode 截取字符串
for($i=0;$i<count($temp);$i++)             //循环显示返回的数组元素
echo $temp[$i]."，";
echo "<P>";
echo "字符串".$s;
echo "<br>";
echo "使用 split 方法分割开：";
$temp=split('[|]',$s);                     //分割字符定义为“|”来取字符串
for($i=0;$i<count($temp);$i++)             //循环显示返回的元素
echo $temp[$i]."，";
echo "<P>";
echo "字符串".$s;
echo "<br>";
echo "使用 strtok 方法分割开：";
$temp=strtok($s,$s2);                      //使用 strtok 截取字符串
while($temp)                               //循环显示分割后的字符串
```

```
{
    echo $temp."，";
    $temp=strtok("|");
}
?>
</body>
</html>
```

保存以上代码为“7-15.php”。在 PHP 执行环境中运行以上代码，将会出现如图 7.15 所示的执行结果。

从以上代码的执行情况可以看出：split()函数需要使用正则表达式规则；而 strtok()函数返回的只是字符串而不是字符串数组；而使用 explode()函数切开字符串是 3 个函数中最简单易用的。



图 7.15　切开字符串使用实例执行结果

7.3.5　去除字符串中的特殊符号

在进行实际 PHP 编程的时候，处理的字符串有可能是来自于网页中，里面可能包含有 HTML 或者 PHP 标记。这时就需要把它们去除之后再做处理。在 PHP 中也有专门除去字符串中特殊符号标记的字符串处理函数：strip_tags()。

strip_tags(string str)函数，去掉字符串参数 str 中的 HTML 及 PHP 标记，将处理之后的字符串作为函数的返回值。

此函数可去掉字符串中包含的任何 HTML 及 PHP 的标记字串。若是字串的 HTML 及 PHP 标签原来就有错，如少了大于号，则也会传回错误。而本函数和文件操作系列的 fgetss()函数有着相同的功能。

下面通过一个实例来说明，如何使用 strip_tags()函数来去除字符串中的特殊标记。

```
<html>
<head>
<title>去除字符串的特殊符号使用实例</title>
</head>
<body>
<?
$s="<font color=\"#ff0000\">我爱北京天安门！</font>";
$t=strip_tags($s);
$s2="<font size=\"16pt\">天安门上太阳升！</font>";
$t2=strip_tags($s2);
echo $s;
echo "<p>";
echo $t;
echo "<p>";
echo $s2;
echo "<p>";
echo $t2;
?>
</body>
</html>
```

保存以上代码为“7-16.php”。在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.16 所示。

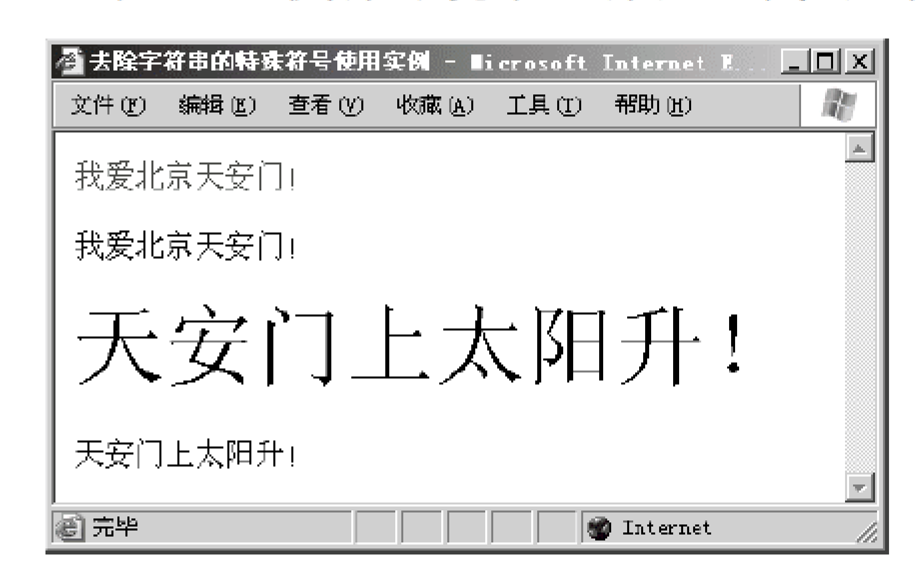


图 7.16　去除字符串的特殊符号使用实例执行结果

通过上例发现，使用 strip_tags()函数去除了原字符串中的字体颜色和字体大小的标识。达到了去除了特殊符号的目的。

7.3.6　转换字符串中的特殊符号为 HTML 标记

编写 Web 互动网页时，安全问题是一定要考虑的。一个不可忽略的问题就是对用户提交信息的处理。如果用户提交了恶意 HTML 代码，就会影响到其他用户的使用，甚至会造成系统瘫痪。所以编写互动 Web 网页时，出于安全考虑要对用户提交内容中的 HTML 代码进行处理。

在 PHP 中有一个函数实现这样的功能：把用户提交内容（字符串）中的特殊符号转换为 HTML 标记。这个函数就是 htmlspecialchars()。

htmlspecialchars(string string)函数，实现功能：将字符串参数 String 中的特殊符号（如<、>、"等）转化为 HTML 标记。其具体转换内容如下：

- &，转换成&。
- "，转换成"。
- <，转换成<。
- >，转换成>。

下面通过一个实例来说明，如何使用 htmlspecialchars()函数对字符串进行转换处理。

```
<html>
<head>
<title>转换字符串的特殊符号为 HTML 标记使用实例</title>
</head>
<body>
<?
$s="<font color=\"#ff0000\">我爱北京天安门！</font>";     //定义第 1 个字符串
$t= htmlspecialchars ($s);                                //对第 1 个字符串进行处理
$s2="<font size=\"16pt\">天安门上太阳升！</font>";         //定义第 2 个字符串
$t2=htmlspecialchars($s2);                                //对第 2 个字符串进行处理
//分别输出原字符串及处理过的字符串以查看处理的结果
echo $s;
echo "<p>";
echo $t;
```

```
echo "<p>";
echo $s2;
echo "<p>";
echo $t2;
?>
</body>
</html>
```

保存以上代码为“7-17.php”。在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.17 所示。

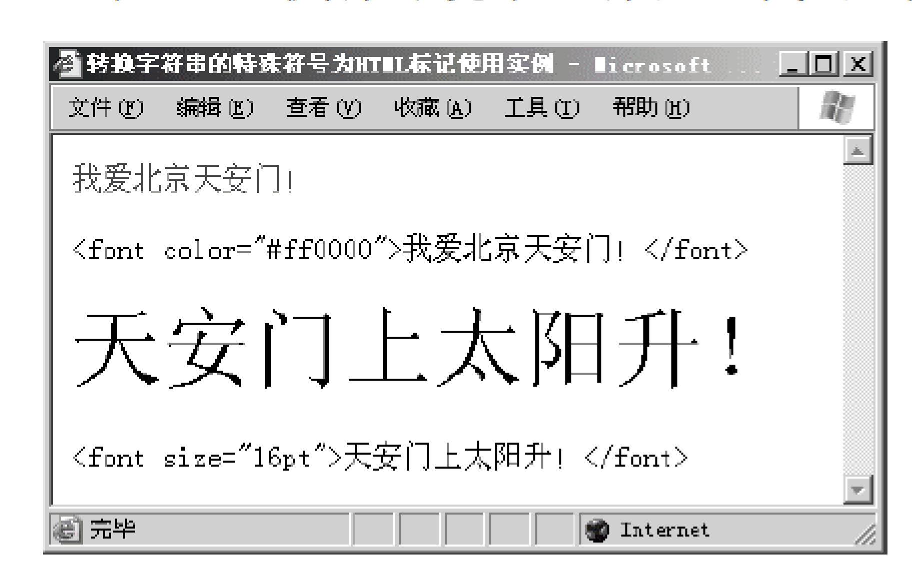


图 7.17　转换字符串的特殊符号为 HTML 标记使用实例执行结果

通过以上函数可见 htmlspecialchars()函数把字符串中的特殊符号都转换成了 HTML 标记。经过这样的处理，加入字符串中的 HTML 代码都将因被转化而不能起作用，这样在客观上就起到了安全的效果。

7.3.7　加入去掉斜线

在数据库操作时，如果把未经操作的单引号写入库里就会使 SQL 语句发生错误。所以给特殊字符加上斜线就显得尤为重要。同理，显示的时候就要把入库时加入的反斜线去除，这样库里的内容才能正常显示。加入斜线 addslashes(string str)函数和去除斜线 stripslashes(string str)函数在 PHP 里也是经常用到的函数。

这两个函数的使用与用户的 PHP 解释器的设置有关。php.ini 是 PHP 设置的核心文件。其中的 magic_quotes_gpc 项的开关对这两个函数有着直接的影响。通常有以下两种情况：

- magic_quotes_gpc=on：可以不对输入和输出数据库的字符串数据作 addslashes()和 stripslashes()的操作，数据也会正常显示。如果此时对输入的数据作了 addslashes()处理，那么在输出的时候就必须使用 stripslashes()去掉多余的反斜杠。
- magic_quotes_gpc=off：必须使用 addslashes()对输入数据进行处理，但并不需要使用 stripslashes()格式化输出，因为 addslashes()并未将反斜杠一起写入数据库，只是帮助 MySQL 完成了 SQL 语句的执行。

由于这两个函数牵涉数据库的操作，这一节只给出简单提示，具体数据库的操作，请参看本书第 14 章使用 MySQL 数据库。

7.3.8　两个字符串的比较

两个数值，不论是整型的还是浮点型的都能够比较大小。而两个字符串也可以比较大小。在 PHP

中，有一个函数专门用来比较字符串的大小。这个函数就是 strcmp()。其使用格式如下：

```
int strcmp(string str1,string str2)
```

参数 str1 与 str2 表示两个字符串变量，函数返回一个整数。如果 str1 大于 str2 则返回正数；如果 str1 小于 str2 则返回负数；如果两个字符串完全一致即字符串相等则返回 0。

字符串大小的规则是：从左向右比较，比较同一位置的字符，ASCII 码表中后面的大于前面的，即 9 大于 1、T 大于 A。

下面用一个实例来说明如何使用 strcmp()函数。

```
<html>
<head>
<title>比较字符串的大小函数使用实例</title>
</head>
<body>
<?
$s="abcdefg";                               //定义变量
$s2="abddefg";
$s3="abddefg";
echo $s;
echo "<p>";
echo $s2;
echo "<p>";
echo $s3;
echo "<p>";
function bijiao($str1,$str2)                //基于 strcmp()函数，构造一个函数
{
    if(strcmp($str1,$str2)>0)               //如果第 1 个字符串大于第 2 个即返回正数
    echo $str1."大于".$str2;
    elseif(strcmp($str1,$str2)<0)           //如果第 1 个字符串小于第 2 个即返回负数
    echo $str1."小于".$str2;
    else                                    //如果两个字符串相等即返回 0
    echo $str1."等于".$str2;
}                                           //bijiao 函数结束
bijiao($s,$s2);                             //调用自定义函数 bijiao
echo "<p>";
bijiao($s2,$s);
echo "<p>";
bijiao($s3,$s2);
echo "<p>";
?>
</body>
</html>
```

保存以上代码为“7-18.php”，然后在 PHP 执行环境中运行以上代码，其执行结果如图 7.18 所示。

从以上执行结果可见返回结果符合比较规则。通过这个例子，读者既能明白 strcmp()函数的使用方法，同时也巩固了自定义函数使用的知识。



图 7.18 比较字符串的大小函数使用实例执行结果

7.3.9 改变字符串的大小写

在进行字符串处理的时候，有时需要对字符串中字母的大小写进行转换。这时就需要用到 strtolower()函数和 strtoupper()函数。

- string strtolower(string str)函数，把字符串参数 str 中的所有大写字母转换为小写字母，把转换后的新字符串作为函数的返回值。
- string strtoupper(string str)函数，把字符串参数 str 中的所有小写字母转换为大写字母，把转换后的新字符串作为函数的返回值。

下面通过一个实例来说明如何使用这两个改变字符串大小写的函数。

```
<html>
<head>
<title>改变字符串的大小写函数使用实例</title>
</head>
<body>
<?
$s="I Love My Great Country!";                    //定义变量
$temp1=strtolower($s);                            //转换为小写字母
$temp2=strtoupper($s);                            //转换为大写字母
echo "原字符为：";
echo $s;
echo "<p>";
echo "原字符经过 strtolower()处理后的值为：";
echo "<br>";
echo $temp1;                                      //打印结果
echo "<p>";
echo "原字符经过 strtoupper()处理后的值为：";
echo "<br>";
echo $temp2;                                      //打印结果
?>
</body>
</html>
```

保存以上代码为“7-19.php”。在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.19 所示。



图 7.19　改变字符串的大小写函数使用实例执行结果

查看以上执行结果发现：经过 strtolower()函数处理的字符串内容全变为小写；而经过 strtoupper()函数处理的字符串内容全变成了大写。从这个例子及执行结果能清晰地了解这两个函数的执行结果。

7.3.10　其他常用字符串函数

除了以上几小节介绍的字符串处理函数之外，在进行 PHP 编程及字符串处理过程中，还有其他一些常用的字符串函数。由于相对于前几小节介绍的函数，这些函数使用频率相对较低。所以只对它们的使用格式、参数、返回值等情况做下简单介绍。

- string chop(string str)函数，去除字符串 str 中的连续空白，返回值为处理后的字符串。
- string ltrim(string str)函数，功能与 chop 类似，也是去除字符串中的连续空白带（whitespace），并把处理结果返回。
- string md5(string str)函数，把字符串 str 进行 MD5 加密，并把加密后的字符串作为函数的返回值。这个函数在处理用户的密码时经常用到，一般是把用户密码经 md5()函数加密后再入库。
- string nl2br(string str)函数，把字符串 str 中的回车换行转换为 HTML 标记中的
，并把处理结果返回。这也是一个很有用的函数，特别是用在用户提交的内容中存在换行时，使用这个函数就能保持用户输入的格式。
- string str_replace(string needle, string str, string haystack)函数，函数将字符串 str 代入 haystack 字符串中，将所有的 needle 置换成 str。例如，使用 str_replace("a","b","abcd")，函数将返回 bbcd。

到这里 PHP 中常用的字符串处理函数就讲解完了。更多的是需要平常的使用，只有经常使用才能熟练掌握这些函数。

7.4　PHP 中常用的数组函数

在 PHP 编程过程中，有相当大的成分是与数组打交道。入库的内容需要纳入数组、统计相关内容要写入数组等都离不开数组，很多的信息都是用数组作为载体的。所以数组的操作在 PHP 编程中占有很大的比重，只有熟练的操作数组才能熟练编写 PHP 程序。这一节就来介绍 PHP 中常用的数组函数。

7.4.1　新建一个数组

既然要操作数组，第 1 步就是要新建一个数组。新建数组也有相关的函数，这个函数就是 array()。从前面对数组的一些介绍中，我们知道，新建一个数组通常有两种方法：一种是直接给数组每个变量赋值；另一种就是同时给数组所有元素赋值。而给数组所有元素同时赋值时就要用到 array()函数。

array array(…)函数，新建一个数列，传回的参数是数列形态。参数可以是带有“=>”运算符的索引。从实质上来说 array()并不算是一个正规的函数，它主要是用来表示数组。

下面通过一个例子来说明一下 array()函数的使用，同时也复习一下如何新建一个数组。

```
<html>
<head>
<title>array()函数使用实例</title>
</head>
<body>
<?
$temp=array(1,2,3,4,5,6);                    //定义数组$temp
$temp1=array(0=>"zero",
1=>"one",
2=>"two",
3=>"three"
);                                           //定义数组$temp1
$temp2=array(name=>"张三",
sex=>"男",
age=>"20"
);                                           //定义数组$temp2
echo $temp[0];                               //输出数组
echo "、";
echo $temp[1];
echo "、";
echo $temp[2];
echo "、";
echo $temp[3];
echo "、";
echo $temp[4];
echo "、";
echo $temp[5];
echo "、";
echo $temp[6];
echo "<p>";
for($i=0;$i<count($temp1);$i++)              //循环输出数组
echo $temp1[$i]."，";
echo "<p>";
echo $temp2[name];                           //输出数组
echo "<p>";
echo $temp2[sex];
echo "<p>";
```

```
echo $temp2[age];
?>
</body>
</html>
```

保存以上代码为“7-20.php”。在 PHP 执行环境中运行以上代码，执行结果如图 7.20 所示。

通过以上实例，不仅使读者了解了 array()函数的使用方法，更加深了对如何建立一个数组的理解。

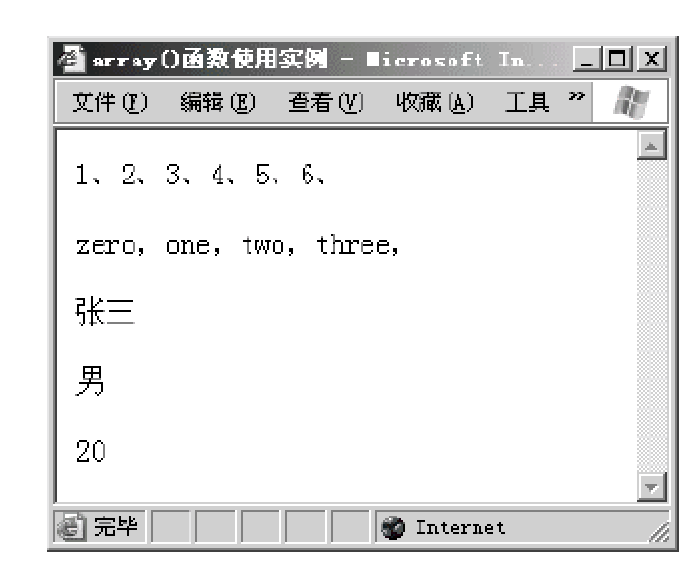


图 7.20　array()函数使用实例执行结果

7.4.2　计算数组的元素个数

在对一个数组操作之前，知道数组的元素个数是很有必要的。如要对数组循环输出，只有当知道了数组的元素个数时才能进行。在 PHP 中计算数组元素个数的方法非常简单，那就是使用 count()函数。

int count(mixed var)函数，参数可以是数组或者整数变量。如果参数是数组则传回数组元素的个数；如果参数为整数则分两种情况：如果整数变量还没有值则传回“0”，如果已经赋值则传回“1”。

下面通过一个实例来说明 count()函数的使用方法。

```
<html>
<head>
<title>计算数组的元素个数实例</title>
</head>
<body>
<?
$arr1[0]="zero";                                        //定义数组变量 1
$arr1[1]="one";
$arr1[2]="two";
$arr1[3]="three";
$arr1[4]="four";
$arr1[5]="five";
$arr1[6]="six";
$arr2=array("中国","美国","俄罗斯","英国","法国");        //定义数组变量 2
$i=5;                                                   //定义变量$i 并赋值
$j;                                                     //定义变量$j
echo "数组 arr1 的元素个数为：";
echo count($arr1);                                      //计算数组 1 元素个数
echo "<p>";
echo "数组 arr2 的元素个数为：";
echo count($arr2);                                      //计算数组 2 元素个数
echo "<p>";
echo count($i);                                         //返回变量值的情况
echo "<p>";
echo count($j);                                         //返回变量值的情况
?>
</body>
</html>
```

保存以上代码为“7-21.php”。在 PHP 执行环境中运行以上代码，看到执行结果如图 7.21 所示。

在 PHP 中还有一个函数也可以完成类似的工作，这个函数是 sizeof()。该函数也是把数组作为函数的参数，把数组元素的个数作为函数的返回值。因为这两个函数使用方法相同，所以这里就不再做过多介绍。

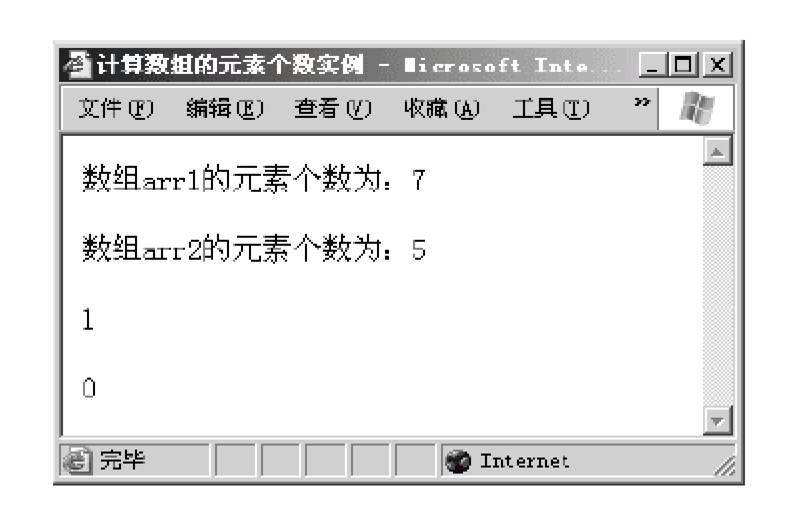


图 7.21　计算数组的元素个数实例执行结果

7.4.3　对数组排序及逆排序

在实际 PHP 编程时，有时需要对数组进行排序。通常对数组进行排序的方法有以下几种：冒泡法、对分法等。但是在 PHP 中对数组的排序就变得相当的简单，因为在 PHP 的数组操作函数中，有专门对数组进行排序的函数。使用该类函数可以对数组进行顺序或逆序排列。这两个函数就是 sort()函数和 rsort()函数。分别介绍如下：

- void sort(array array)函数，对数组进行排序，使数组按照从小到大的顺序重新排列。
- void rsort(array arry)函数，和 sort()函数一样对数组进行排序，与 sort()函数不同的是 rsort()函数将使数组按从大到小的顺序重新排列。

下面通过一个实例来说明，如何使用这两个函数实现对数组进行排序操作。具体代码如下：

```
<html>
<head>
<title>对数组排序函数使用实例</title>
</head>
<body>
<?
$temp=array(5,2,6,4,1,3);                              //定义数组$temp
$temp1=array("北京","南京","上海","杭州","重庆");      //定义数组$temp1
echo "数组 temp 原始顺序为：";
for($i=0;$i<count($temp);$i++)                         //通过循环打印原始数组
{
    echo $temp[$i]."，";
}
echo "<p>";
echo "数组 temp 经 sort 函数处理后的顺序为：";
sort($temp);                                           //对$temp 数组进行 sort 处理
for($i=0;$i<count($temp);$i++)                         //通过循环打印处理过的数组
{
    echo $temp[$i];
    echo "，";
}
echo "<p>";
echo "数组 temp 经 rsort 函数处理后的顺序为：";
rsort($temp);                                          //对$temp 数组进行 rsort 处理
for($i=0;$i<count($temp);$i++)                         //通过循环打印处理过的数组
{
    echo $temp[$i];
```

```
        echo "，";
}
echo "<p>";
echo "------------------------------------------------";
echo "<p>";
echo "数组 temp1 原始顺序为：";
for($i=0;$i<count($temp1);$i++)                          //通过循环打印原始数组
{
        echo $temp1[$i];
        echo "，";
}
echo "<p>";
echo "数组 temp1 经 sort 函数处理后的顺序为：";
sort($temp1);                                            //对$temp 数组进行 sort 处理
for($i=0;$i<count($temp1);$i++)                          //通过循环打印处理过的数组
{
        echo $temp1[$i];
        echo "，";
}
echo "<p>";
echo "数组 temp1 经 rsort 函数处理后的顺序为：";
rsort($temp1);                                           //对$temp 数组进行 rsort 处理
for($i=0;$i<count($temp1);$i++)                          //通过循环打印处理过的数组
{
        echo $temp1[$i];
        echo "，";
}
?>
</body>
</html>
```

保存以上代码为“7-22.php”。在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.22 所示。

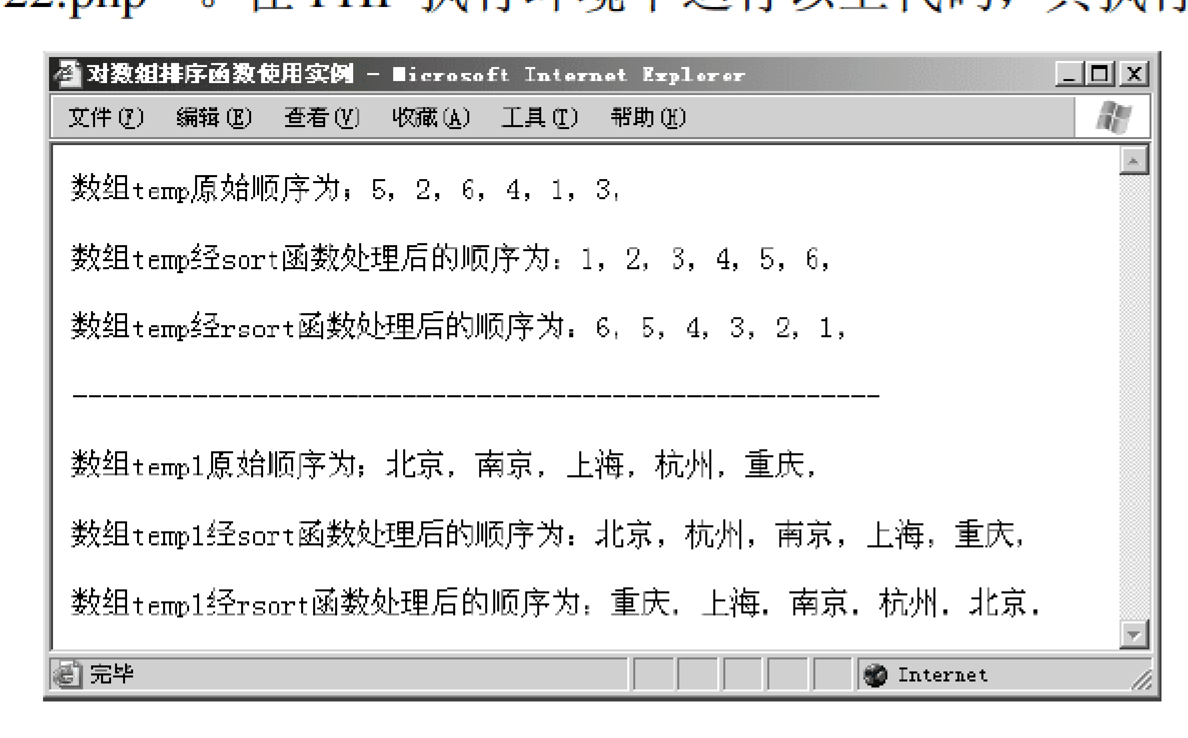


图 7-22　对数组排序函数使用实例执行结果

通过函数执行结果可以发现，在 PHP 中对数组进行排序是一件很简单的事情。

7.4.4　对数组进行自定义排序

有的时候，单纯使用 7.4.3 小节介绍的 sort()函数或 rsort()函数均不能完全满足对数组排序的要求。如下面的数组：

```
<?
$temp=array("班长","无职务","副班长","团支书");
?>
```

如果想要实现这样的排序：团支书>班长>副班长>无职务。这时不管是使用 sort()函数还是使用 rsort()函数，都不好使。因为这种排序标准是自定义的。出现这种情况，就要使用另一个 PHP 数组操作函数 usort()了。

void usort(array array,function cmp_function)函数，对数组进行排序，使数组按照用户自定义比较函数所规定的顺序重新排列。

下面通过一个实例来说明 usort()函数的使用方法，通过这个例子来解决上面提到的问题。

```
<html>
<head>
<title>对数组自定义排序函数使用实例</title>
</head>
<body>
<?
$temp=array("班长","无职务","副班长","团支书","无职务","副班长");   //定义数组$temp
function cmp($a,$b)                                                 //自定义排序函数
{
        if($a=="团支书")                                            //第 1 个参数为团支书的情况
        {
                if($b=="团支书")
                return 0;
                elseif($b=="班长")
                return -1;
                elseif($b=="副班长")
                return -1;
                else
                return -1;
        }
        elseif($a=="班长")                                          //第 1 个参数为班长的情况
        {
                if($b=="团支书")
                return 1;
                else if($b=="班长")
                return 0;
                else if($b=="副班长")
                return -1;
                else
                return -1;
```

```
        }
        elseif($a=="副班长")                                    //第 1 个参数为副班长的情况
        {
            if($b=="团支书")
            return 1;
            else if($b=="班长")
            return 1;
            else if($b=="副班长")
            return 0;
            else
            return -1;
        }
        else                                                    //第 1 个参数为无职务的情况
        {
            if($b=="团支书")
            return 1;
            else if($b=="班长")
            return 1;
            else if($b=="副班长")
            return 1;
            else
            return 0;
        }
    }                                                           //根据不同情况返回不同的值
    echo "数组 temp 原始顺序为：";
    echo "<p>";
    for($i=0;$i<count($temp);$i++)                              //通过循环打印原始数组
    {
        echo $temp[$i]."，";
    }
    usort($temp,"cmp");                                         //对数组进行 usort 处理
    echo "<p>";
    echo "数组 temp 经过 usort 处理过的顺序为：";
    echo "<p>";
    for($i=0;$i<count($temp);$i++)                              //通过循环打印处理过的数组
    {
        echo $temp[$i]."，";
    }
    ?>
    </body>
    </html>
```

保存以上代码为“7-23.php”。在 PHP 执行环境中运行以上代码，将会看到如图 7.23 所示的执行结果。

通过以上实例及执行结果可见，使用 usort()函数，可以实现一些自定义的排序规则，从而实现对数组元素进行自定义排序。

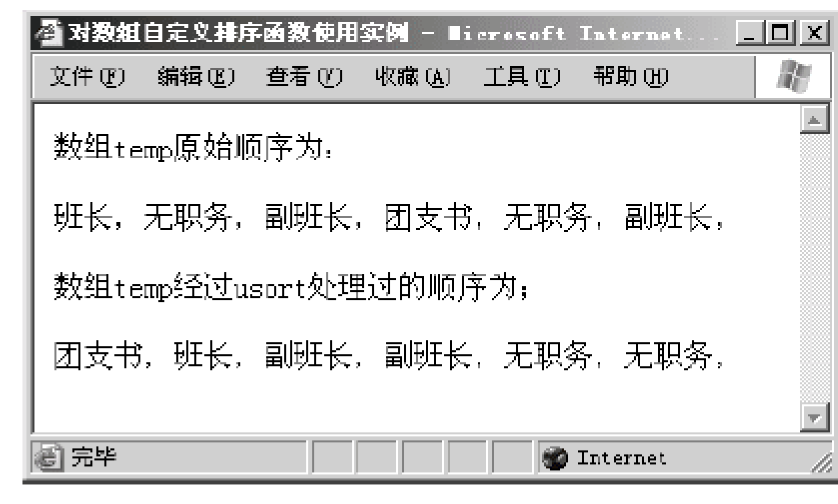


图 7.23　对数组自定义排序函数使用实例执行结果

7.4.5　移动数组指针

经过 7.4.4 小节的介绍，读者了解到数组有很多元素，那么究竟数组当前元素是哪一位？这里就需要指出一个数组指针的问题。每一个数组变量都有一个内部指针，它指向当前的数组元素。在进行 PHP 编程中，有时需要对数组的指针进行移动操作。在 PHP 库函数中，有一组函数来实现这一操作。这些函数有以下几种：

- current()函数，传回数组当前指针指向的元素。
- end()函数，将数组的指针移动到数组尾部，即指向数组最后的元素。
- next()函数，将数组的指针向后移动一位，即指向当前的后一个元素。
- prev()函数，将数组的指针向前移动一位，即指向当前的前一个元素。
- reset()函数，将数组的指针移动到数组头部，即指向数组的第一个元素。

下面通过一个实例来向读者介绍，如何使用这些函数来达到移动数组指针的目的。具体内容参看以下代码：

```
<html>
<head>
<title>移动数组指针函数使用实例</title>
</head>
<body>
<?
$new[0]="zero";                                    //定义一个数组以备用
$new[1]="one";
$new[2]="two";
$new[3]="three";
$new[4]="four";
$new[5]="five";
$new[6]="six";
for($i=0;$i<count($new);$i++)                      //通过循环打印数组
{
     echo $new[$i]."，";
}
echo "<p>";
echo "指针当前指向：";
echo current($new);                                //打印当前数组元素，以确定指针位置
echo "<p>";
echo "指针后移一位：";
next($new);                                        //数组指针后移一位
echo current($new);                                //查看当前数组指针指向
echo "<p>";
echo "指针移动尾部：";
end($new);                                         //把数组指针移动的数组尾部
echo current($new);                                //查看是否已到尾部
echo "<p>";
echo "指针前移一位：";
```

```
prev($new);                                 //把数组指针前移一位
echo current($new);                         //查看指针指向情况
echo "<p>";
echo "指针移动到头部：";
reset($new);                                //把数组指针指向头部
echo current($new);                         //查看是否已到头部
?>
</body>
</html>
```

把以上代码保存为“7-24.php”。在 PHP 执行环境中运行该文件，执行结果如图 7.24 所示。

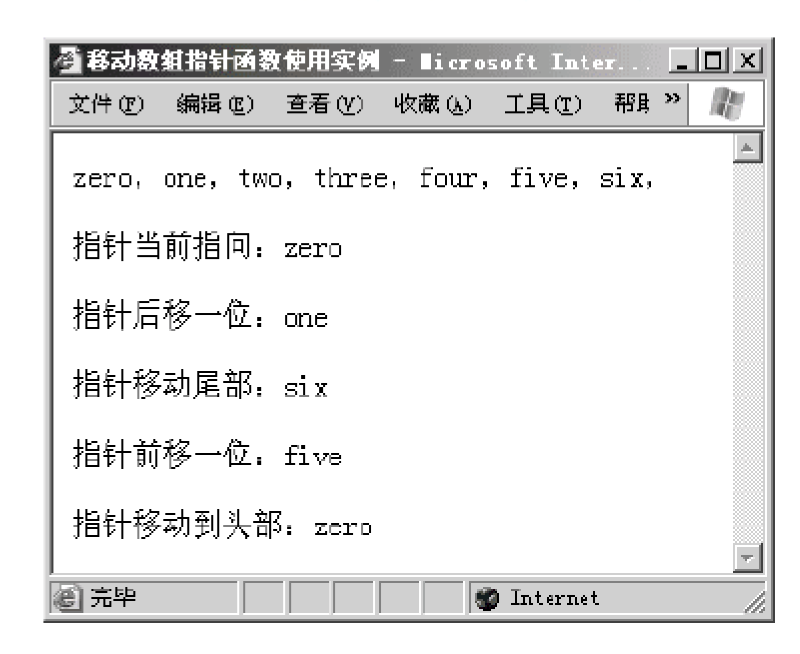


图 7.24　移动数组指针函数使用实例执行结果

下面对以上代码及执行结果做一下简单分析，首先初始化数组，数组指针指向数组第 1 个元素，所以 current()返回数组的第 1 个元素 zero；指针后移一位则指向 one；指针移动到数组的尾部则指向最后一个元素 six；指针前移一位则指向 five；指针移动到头部则指向 zero。

7.4.6　获取数组当前元素

通过 7.4.5 小节对 PHP 中移动数组指针函数的介绍，读者知道要获得数组当前元素，可以使用的方法就是使用 current()函数。

mixed current(array array)函数的返回值即为当前数组指针指向的元素。另外，还有一个 pos()函数，此函数实际上就是 current()函数，两个函数具有相同的参数、作用及返回值。通过两个函数都可以获得数组当前的元素。

7.4.7　移去数组中重复的值

如果一个数组中有大量的数据，在进行数组处理的时候，将其重复冗余数据的值移去就显得很有必要。因为这样可以减少数据量，从而加快数据的处理速度。在 PHP 中有一个函数其作用就是移去数组中重复的值，这个函数就是 array_unique()。

array array_unique(array array)函数，移去数组中重复的值。将数组参数 array 中重复的值移去，将处理过的新数组作为函数的返回值返回。下面通过一个实例来向大家说明 array_unique()函数的使用方法。具体内容参见以下代码：

```
<html>
<head>
<title>移去数组中重复的值函数使用实例</title>
</head>
<body>
<?
$new[0]="zero";                          //定义一个数组以备用
$new[1]="one";
$new[2]="zero";
$new[3]="three";
$new[4]="zero";
$new[5]="five";
$new[6]="six";
for($i=0;$i<count($new);$i++)            //通过循环打印数组
{
    echo $new[$i].", ";
}
echo "<p>";
$new1=array_unique($new);                //对数组进行去除重复值处理
for($i=0;$i<count($new1);$i++)           //通过循环打印新数组
{
    echo $new1[$i].", ";
}
?>
</body>
</html>
```

保存以上代码为“7-25.php”。在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.25 所示。

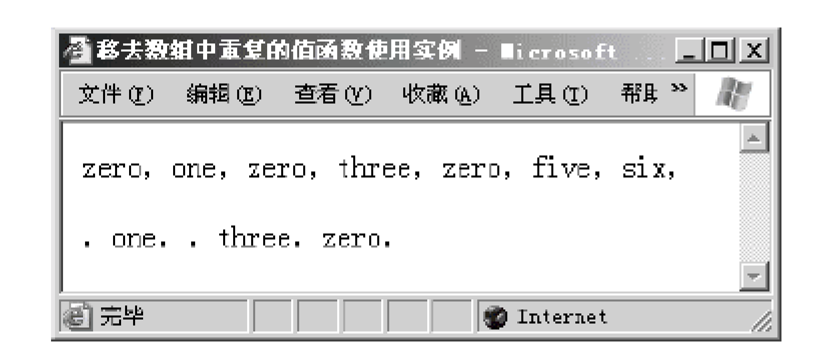


图 7.25　移去数组中重复的值函数使用实例执行结果

查看以上代码及执行结果可见，使用 array_unique()函数对数组进行操作，移去了数组中重复的值。

在使用 array_unique()函数时有一点是需要注意的，即虽然去除了数组中的重复值，但数组的键名仍将保持不变。从以上执行结果中就可以看出，$new1[0]、$new1[2]的值为空值但其键名仍在。

7.4.8　计算数组中所有值出现的次数

在使用数组时，如果一个数组中有一个或几个重复的值多次出现，那么统计数组中值的出现次数也是很有必要的。如将一个班级所有学生的年龄赋值给数组，计算某个年龄的学生有几人的情况。这时就要用到 PHP 数组函数中的 array_count_values()函数。

array array_count_values(array array)函数，统计数组中所有值出现的次数。此函数返回一个数组，

该数组用参数数组中的值作为键名，用参数数组中该值出现的次数作为值。

下面通过一个实例来说明此函数的使用方法。具体内容参见以下代码：

```
<html>
<head>
<title>计算数组中所有值出现的次数函数使用实例</title>
</head>
<body>
<?
$new[0]="zero";                              //定义一个数组以备用
$new[1]="one";
$new[2]="zero";
$new[3]="zero";
$new[4]="one";
$new[5]="two";
$new[6]="zero";
for($i=0;$i<count($new);$i++)                //通过循环打印数组
{
	echo $new[$i].", ";
}
echo "<p>";
$new1=array_count_values($new);              //使用 array_count_values()进行处理
echo "zero 出现的次数为：";
echo $new1[zero];                            //打印 zero 出现的次数
echo "<p>";
echo "one 出现的次数为：";
echo $new1[one];                             //打印 one 出现的次数
echo "<p>";
echo "two 出现的次数为：";
echo $new1[two];                             //打印 two 出现的次数
?>
</body>
</html>
```

保存以上代码为“7-26.php”。在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.26 所示。

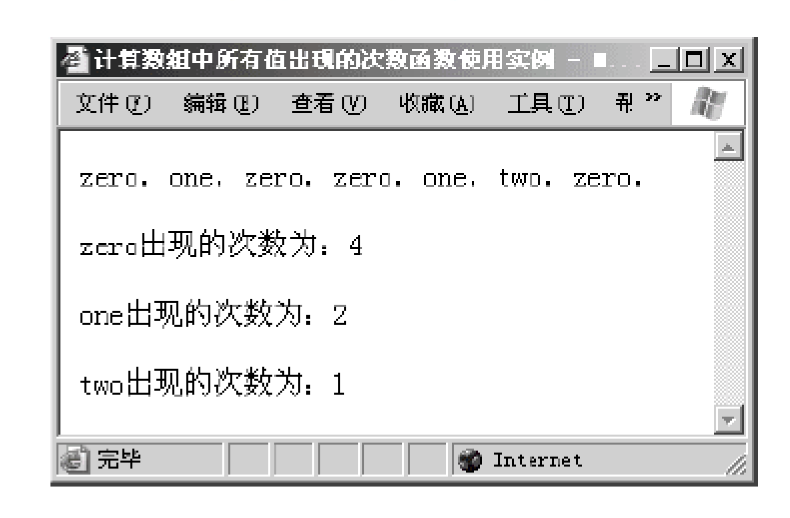


图 7.26　计算数组中所有值出现的次数函数使用实例执行结果

查看以上代码及执行结果可见，通过 array_count_values()函数，返回了原数组中的值及值出现的次数作为新数组，从而实现了对数组中值出现次数的统计。

7.4.9　合并多个数组

实际进行 PHP 编程时，对多个相关的或者类型相同的数组进行操作，其复杂程度要远比对一个数组进行操作大。所以如果能把多个数组合并为一个数组，就能起到简化操作的目的。库函数功能丰富的 PHP 编程环境也为用户准备了这样的函数，这个函数就是 array_merge()。

array array_merge(array array1,array array2,array array3……)函数，此函数将对作为函数参数的多个数组进行合并。一个数组中的值附加在前一个数组后面，把合并后的新数组作为函数的返回值。

在使用此函数时有一点需要注意，如果两个数组中存在相同的字符键名，那么后一个数组中的同键名的值，将替换前一个数组中相应元素的值。代码如下：

```
$a=array(a=>"a",b=>"b",c=>"c");
$b=array(a=>"d",b=>"e",c=>"f");
```

对以上两个数组进行合并，数组$b 中的元素$b[a]、$b[b]、$b[c]将替代数组$a 中相应的元素。

如果是相同的数字键值，并不会出现这样的操作，即后一个数组的值替换前一个数组中的值。代码如下：

```
$a=array("a","b","c");
$b=array("d","e","f");
```

对这两个数组进行合并，因为它们都是以数字作为键值，如$a[0]="a"、$b[0]="d"、……、$a[2]="c"、$b[2]="f"。这样的合并，并不会出现值替代的情况。数组$b 中的元素，将作为新的元素加入到新合并的数组中。

下面通过一个实例来说明 array_merge()函数的使用方法。具体内容参看以下代码：

```
<html>
<head>
<title>合并多个数组函数使用实例</title>
</head>
<body>
<?
$a=array(1,2,3,4,5,6);                          //定义数组$a
$b=array(7,8,9,10,11);                          //定义数组$b
echo "数组 a 的内容为：";
for($i=0;$i<count($a);$i++)                     //循环打印数组$a
{
    echo $a[$i].", ";
}
echo "<p>";
echo "数组 b 的内容为：";
for($i=0;$i<count($b);$i++)                     //循环打印数组$b
{
    echo $b[$i].", ";
}
echo "<p>";
```

```
$c=array_merge($a,$b);                          //对两个数组进行合并
echo "合并后的数组 c 的内容为：";
for($i=0;$i<count($c);$i++)                     //循环打印合并后的数组
{
     echo $c[$i]."，";
}
echo "<p>";
$str1=array(
name=>"张三",
sex=>"男",
length=>"170CM"
);                                              //定义数组$str1
$str2=array(
name=>"李四",
birthday=>"5 月 13 号",
length=>"175CM"
);                                              //定义数组$str2
echo "数组 str1 的内容为：";
print_r($str1);                                 //格式化显示$str1 的内容
echo "<p>";
echo "数组 str2 的内容为：";
print_r($str2);                                 //格式化显示$str2 的内容
$str3=array_merge($str1,$str2);                 //对两个数组进行合并
echo "<p>";
echo "合并后的数组 str3 的内容为：";
print_r($str3);                                 //格式化显示合并后的数组
?>
</body>
</html>
```

保存以上代码为“7-27.php”。然后在 PHP 执行环境中运行以上代码，其执行结果将会如图 7.27 所示。

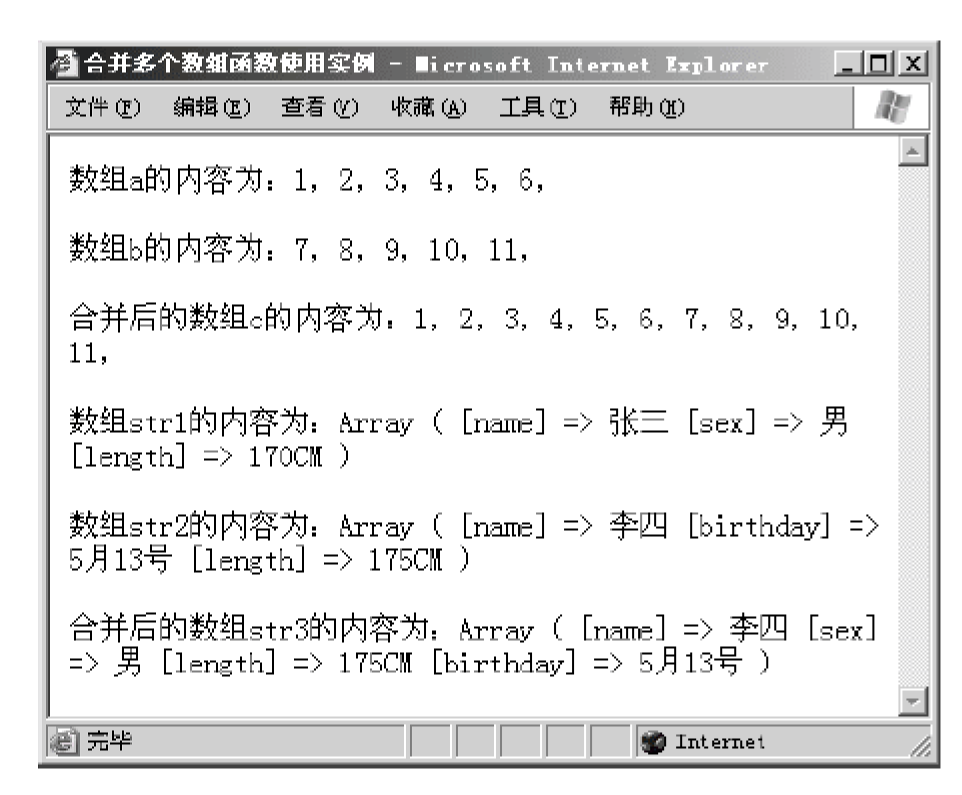


图 7.27　合并多个数组函数使用实例执行结果

查看以上实例及执行结果可见，通过 array_merge()函数，将多个数组进行了合并。并且其遵循这样的规则：新的字符键值的值替换以前的字符键值的值，而数字键值的值只作为新值累加。

7.4.10　其他常用数组函数

在实际使用 PHP 进行网络编程的过程中，除了以上介绍的数组函数之外还有以下常用的 PHP 数组函数。由于这些函数使用频率相对于前几小节介绍的函数较低，所以只作简单介绍而不再举实例了。

- bool in_array(mixed needle, array haystack [, bool strict])函数。在 haystack 中搜索 needle，如果找到则返回 True，否则返回 False。
- bool ksort(array &array [, int sort_flags])函数。对数组的键名进行排序，保留键名到数据的关联。
- bool natcasesort(array &array)函数。用自然排序法（及按照从小到大的顺序进行排序），对数组进行不区分大小写字母的排序。并把排好的结果作为函数返回值。

PHP 中常用的数组函数就介绍到这里，更多的内容请参考相关手册，通过实践来提高能力。

7.5　本 章 小 结

这一章为读者介绍了 PHP 的数据处理，包括 PHP 中常用的数学函数、字符串函数、数组函数等内容。熟练掌握并在实际工作中运用这些函数对数据的操作，将起到事半功倍的效果。数据处理是编写大型 Web 应用程序的基础，只有正确地对数据进行相应的处理，才能为大型应用程序提供基础数据支持。

第 8 章

PHP 文件应用

- 使用 PHP 打开文件
- 使用 PHP 读取文件内容
- 使用 PHP 删除文件
- PHP 对目录的操作
- 关于文件上传

文件操作在 PHP 编程中占有很重要的地位。在很多情况下，用户都要对文件进行操作，如文本计数器、文本留言板、文件管理系统甚至是文本数据库等。本章将详细介绍在 PHP 中如何判断文件是否存在、如何获取文件属性、如何读取文件内容、如何对目录进行操作等文件操作。通过对本章的学习，将使读者对如何使用 PHP 进行文件操作有一个全面的认识。

8.1 判断文件是否存在

在对一个文件进行操作之前，判断该文件是否存在是必要的。如果用不恰当的方法打开一个并不存在的文件，就会导致错误。在PHP中有一个专门的函数来做这项工作，这个函数就是file_exists(string)。参数 string 为一个指向文件或目录的字符型变量，函数的返回值为布尔型变量，如果文件或目录存在，返回值为 True，反之则返回 False。下面用一个简单的例子来说明这个问题。代码如下：

```
<html>
<head>
<title>file_exists()函数使用实例</title>
</head>
<body>
<?                                              //file_exists()使用实例
$filename="data.txt";                           //定义变量
if(file_exists($filename))                      //如果文件存在时执行操作
{
    echo "指定文件".$filename."存在";
}
else                                            //如果文件不存在时执行操作
{
    echo "指定文件".$filename."不存在！";
}
?>
</body>
</html>
```

保存以上代码为“8-1.php”。在 PHP 执行环境中运行该文件，如果在该文件（8-1.php）同名目录下有 data.txt 文件，则会返回“文件存在”；反之，则会返回“文件不存在”。

通过上面的例子，读者可以清晰地认识到 file_exists()函数是如何使用的。

不过，有一个问题需要读者注意，基于安全考虑，大多数的文件操作类函数的文件名参数不能为远程文件，即只能通过服务器的文件系统访问。其中包括本节的 file_exists()函数及 8.2 节将要介绍的函数。例如这样的引用：file_exists("http://www.sohu.com/index.mdb")（其实并不存在该文件），不论文件存在与否都将返回 False 值。

8.2 访问文件属性

在进行编程的时候，需要使用到文件的一些常见属性，如文件的大小、文件的类型、文件的修改时间、文件的访问时间和文件的权限等。通过本节的学习，将会掌握使用 PHP 的相关文件函数。要获得这些属性，就如同<?echo "hello! world"?>一样简单。

具体内容请参考表 8.1。

表 8.1　文件属性相关函数

函　数　名	作　　用	参数及返回值
filesize(string)	获取文件的大小	参数string为一个指向文件或目录的字符型变量。函数的返回值为整型变量，返回文件的大小（字节）。如果出错，则返回False。函数参数不能为远程文件，返回结果会被缓存
filetype(string)	获取文件的类型	参数string为一个指向文件或目录的字符型变量。函数的返回值为字符型变量，可能出现的值有fifo、char、dir、block、link、file和unknown，返回值会被缓存
filemtime(string)	获取文件的修改时间	参数string为一个指向文件或目录的字符型变量。函数的返回值为整型变量，返回文件的修改时间
fileatime(string)	获取文件的访问时间	参数string为一个指向文件或目录的字符型变量。返回值为整型变量，内容为文件的访问时间
fileperms(string)	获取文件的权限	参数string为一个指向文件或目录的字符型变量。返回值为整型变量，内容为文件相应的权限，同其他这一类型的函数一样，参数不能为远程文件，另外返回结果同样会被缓存

下面通过一个简单的例子来实际使用这些函数。注意，在使用这个例子之前，要保证同级目录下有 data.txt 这个文件。代码如下：

```
<html>
<head>
<title>访问文件属性函数使用实例</title>
</head>
<body>
<?                                                      //获取文件属性使用实例
$filename="data.txt";                                   //定义变量
echo $filename."的大小为：".filesize($filename)."<br>";    //使用 filesize()函数
echo $filename."的类型为：".filetype($filename)."<br>";    //使用 filetype()函数
//使用 filemtime()函数并格式化返回日期
echo $filename."的修改时间为：".date("Y 年 n 月 t 日",filemtime($filename))."<br>";
//使用 fileatime()函数并格式化返回日期
echo $filename."的访问时间为：".date("Y 年 n 月 t 日",fileatime($filename))."<br>";
//使用 fileperms()函数
echo $filename."的权限为：".fileperms($filename)."<br>";
?>
</body>
</html>
```

把以上代码保存为“8-2.php”。在 PHP 执行环境中执行以上代码，执行结果如图 8.1 所示（当然会因为文件 data.txt 的不同，得出结果有所不同）。

读者从这个例子中可以看到，使用 PHP 文件相关函数获得文件属性是一件很简单的事情。

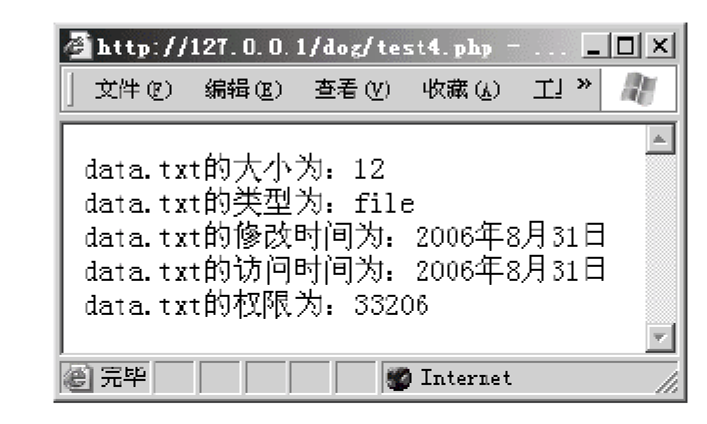


图 8.1　访问文件属性

8.3　打 开 文 件

要使用一个文件，第一件事就是要把文件打开。PHP 中提供了一个很有用的库函数：fopen(string file , string mode)，它的作用就是打开本地或者远程文件。参数 filename 为一个字符型变量，代表想要打开的文件的文件名；参数 mode 为打开模式，可选的参数有 6 个，分别是 r、r+、w、w+、a 和 a+。根据所选参数的不同，又可以分为以下几种不同的打开类型。

8.3.1　用只读方式打开文件

用只读方式打开文件使用：fopen(string file,"r")函数。参数 r 使打开模式为只读，文件指针指向文件开头处。这样打开的文件，不能被写入。下面通过一个实例来说明。代码如下：

```
<html>
<head>
<title>fopen()函数使用实例</title>
</head>
<body>
<?
//fopen()使用实例
$filename="data.txt";                                    //定义变量
//使用 fopen()函数打开文件
//并且使用“r”参数设置打开模式为只读
$myfile=fopen($filename,"r");
fwrite($myfile,"hello world!");                          //试图进行写入操作
fclose($myfile);                                         //关闭打开的文件
?>
</body>
</html>
```

把以上代码保存为“8-3.php”文件。在 PHP 运行环境中执行以上代码，会发现 fwrite()（对文件进行写入操作）这一句并没有被执行。打开文件 data.txt 会发现，里面也并没有多出“hello world!”字符串。这是因为$myfile 句柄的打开方式为只读，所以不能进行写入操作。

8.3.2　用写入方式打开文件

用写入方式打开文件：fopen(string file,"w")和 fopen(string file,"a")。参数 w、a 使打开模式为写入，这样打开的文件能够被写入。另外，使用这两个参数，当文件不存在时，能自动创建文件。两者的不同在于，使用参数 w，文件指针指向文件开始处并将原文件清空（所以这是一个很危险的参数，除非认为原来的内容没用了，否则不要使用此参数）；而使用参数 a 则使文件指针指向文件结尾而且不会清空原文件内容。这里仍通过 8.3.1 小节的实例来说明，不同的是，把其中的 r 替换为 w。

```
<html>
<head>
<title>fopen()函数使用实例 2</title>
</head>
<body>
<?
//fopen()使用实例
$filename="data.txt";                              //定义变量
//使用 fopen()函数打开文件
//并且使用“w”参数设置打开模式为写入
$myfile=fopen($filename,"w");
fwrite($myfile,"hello world!");                    //试图进行写入操作
fgets($myfile,255);                                //进行读取操作
fclose($myfile);                                   //关闭打开的文件
?>
</body>
</html>
```

保存以上代码为“8-4.php”文件。在 PHP 运行环境中执行以上代码，然后打开文件 data.txt，发现里面多出了“hello world!”字符串，则说明 fwrite()这句对文件进行写入操作的语句被执行了。但是以写入方式打开的文件只能被写入而不能通过 fgetc()和 fgets()等函数来读取文件的内容。如上例中的 fgets()语句就没有被顺利执行。

通过这两个例子，相信读者对文件的只读与写入这两种打开方式的不同有了深刻体会。

8.3.3　用读写方式打开文件

用读写方式打开文件：fopen(string file,"r+")、fopen(string file,"w+")和 fopen(string file,"a+")。参数 r+、w+、a+使打开模式为读写，这样打开的文件，既能够被读取也能够被写入。但是 3 个参数也是有所不同的。如下：

- 参数“r+”文件指针指向文件开头。
- 参数“w+”文件指针指向文件末尾并清空原文件。
- 参数“a+”与“w+”基本类似，不同的是它并不清空原文件。

继续用实例来说明它们。代码如下：

```
<html>
<head>
<title>fopen()函数使用实例 3</title>
</head>
<body>
<?
//fopen()使用实例
$filename="data.txt";                              //定义变量
//使用 fopen()函数打开文件
//并且使用“r+”参数设置打开模式为读写
$myfile=fopen($filename,"r+");
fwrite($myfile,"hello world!");                    //试图进行写入操作
```

```
fgets($myfile,255);                          //进行读取操作
fclose($myfile);                             //关闭打开的文件
?>
</body>
</html>
```

把以上代码保存为“8-5.php”。在 PHP 运行环境中执行以上代码，打开页面中显示出了 hello world!。然后打开文件 data.txt 会发现，里面多出了“hello world!”字符串，说明对文件的读写操作均被正确执行。

下面通过表 8.2 来说明这 6 个参数的异同。

表 8.2　文件打开方式的异同

参　　数	r	r+	w	w+	a	a+
读写方式	只读	读写	可写	读写	可写	读写
指针位置	文件头	文件头	文件尾	文件尾	文件尾	文件尾
是否清空原文件内容	否	否	是	是	否	否
文件不存在是否创建	否	否	是	是	否	否

8.4　读取文件内容

打开文件后就要读取文件的内容了。PHP 的文件函数提供了多种方法来完成这一任务。有读取单个字符的 fgetc()函数和读取文件某一行信息的 fgets()函数和读取整个文件内容的 file()函数等。本节将详细介绍在 PHP 编程环境中如何读取文件内容。在介绍本节内容之前，首先在有 PHP 运行权限的目录内建立“data.txt”文件，并输入以下内容：

```
<html>
<head>
<title>读取文件内容</title>
</head>
<body>
读取文件内容
<p>
<u>读取文件内容</u>
</body>
</html>
```

以备后用。

8.4.1　读取文件相应字符

如果只读取一个文件中的某一个字符，使用 fgetc(int fp)是很方便的。该函数的参数 fp 是已经被打开的文件句柄，函数返回当前文件指针所指向的字符。如果文件指针指向文件末尾，则返回 False。这里通过一个例子来说明 fgetc()函数的使用方法。代码如下：

```
<html>
<head>
<title>fgetc()函数使用实例</title>
</head>
<body>
<?
//打开文件的同时读取文件指针指向的字符
$myfile=fopen("data.txt","r");                    //用只读打开文件文件指针指向文件开头
$mychar=fgetc($myfile);                           //用 fgetc()读取文件指针处字符并赋值给变量
echo $mychar;                                     //显示变量
fclose($myfile);                                  //关闭打开的文件
?>
</body>
</html>
```

把以上代码保存为“8-6.php”文件。在 PHP 运行环境中执行以上代码，将返回“<”。因为文件指针指向文件头，指针处内容赋值给变量。下面对以上代码做些改动，通过循环读取文件所有内容。代码如下：

```
<html>
<head>
<title>fgetc()函数使用实例 2</title>
</head>
<body>
<?
//通循环读取文件所有内容
$myfile=fopen("data.txt","r");                    //用只读打开文件文件指针指向文件开头
while(!feof($myfile))                             //通过循环判断指针是否指向文件末尾
{
    $mychar=fgetc($myfile);                       //用 fgetc()读取文件指针处字符并赋值给变量
    echo $mychar;                                 //显示变量
}
fclose($myfile);                                  //关闭打开的文件
?>
</body>
</html>
```

把代码保存为“8-7.php”。执行该 PHP 文件，执行结果如图 8.2 所示。

查看打开页的源文件，会发现内容与 data.txt 完全一样，说明文件内容被正确读取。这里顺便提一下 feof()函数，它的作用就是判断文件指针是否指向文件末尾，如果已经指向文件末尾，返回 True，反之则返回 False。

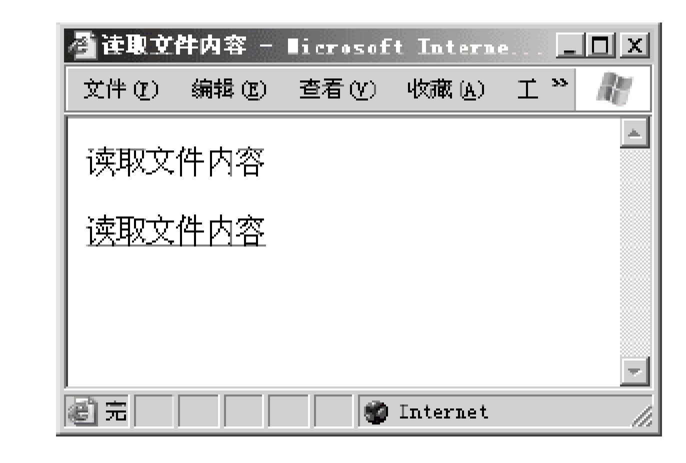


图 8.2　通过循环使用 fgetc()读取文件内容

8.4.2　按行返回文件内容

如果需要按行读取文件内容，使用 fgets(int fp,int)函数。和 fgetc()函数一样，fp 参数是已经被打开

的文件句柄，第 2 个 int 参数为要读取字符的个数。函数返回当前文件指针所指向行指定的字符个数。如果文件指针指向文件末尾，则返回 False。下面通过一个例子来说明该函数的使用方法。代码如下：

```
<html>
<head>
<title>fgets()函数使用实例</title>
</head>
<body>
<?
//打开文件的同时读取文件指针指向的行
$myfile=fopen("data.txt","r");                    //用只读打开文件，文件指针指向文件开头
$myline=fgets($myfile,255);                       //用 fgets()读取文件指针处行并赋值给变量
echo $myline;                                     //显示变量
fclose($myfile);                                  //关闭打开的文件
?>
</body>
</html>
```

保存上面代码为“8-8.php”。在 PHP 运行环境中执行以上代码，会发现什么也没有，但查看源文件会发现“<html>”。因为文件指针指向文件头，指针指向的行为第一行，即“<html>”赋值给了变量。同 fgetc()一样，fgets()也可以通过循环来实现显示文件全部内容，这里就不再赘述。

8.4.3　按行返回文件内容去除 HTML 标记

有时返回文件指定行的内容同时需要去除掉 html 标记，这时使用 fgetss(int fp,int)函数就很方便了。函数使用方法与 fgets()相同，不同的是函数返回行内容的同时去除 html 标记。同样使用例子来说明这个函数的使用。这次直接使用循环来读取文件所有内容。代码如下：

```
<html>
<head>
<title>fgetss()函数使用实例</title>
</head>
<body>
<?
//打开文件的同时读取文件指针指向的行
$myfile=fopen("data.txt","r");                    //用只读打开文件文件指针指向文件开头
while(!feof($myfile))
{
$myline=fgetss($myfile,255);                      //用 fgetss()读取文件指针处行并值赋给变量
echo $myline;                                     //显示变量
}
fclose($myfile);                                  //关闭打开的文件
?>
</body>
</html>
```

把以上代码保存为“8-9.php”。在 PHP 运行环境中执行以上代码，将会出现如图 8.3 所示的执行

结果。

查看源文件，只有以下内容：

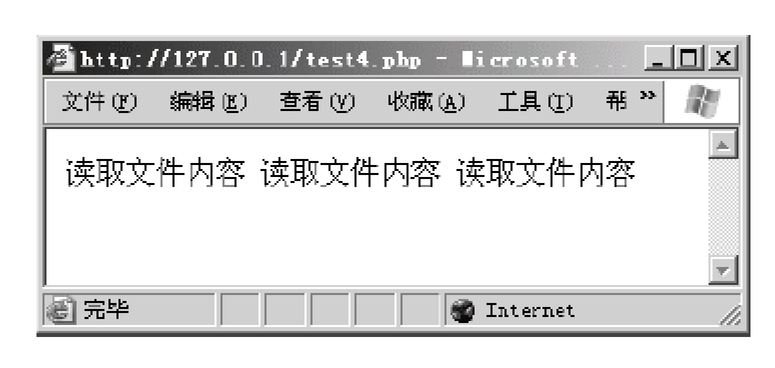


图 8.3　通过循环使用 fgetss()读取文件内容

```
读取文件内容


读取文件内容

读取文件内容
```

之所以这样，是因为 data.txt 文件只有第 3、第 6 和第 8 行含有文字内容，其他行均为 html 标记。在使用 fgetss()函数去除标记后，也就只剩第 3、第 6 和第 8 行被输出了。

8.4.4　将整个文件内容读入到数组变量中

通常把一个多行的文件的全部内容读入到数组变量中会使用 file(string filename)函数。参数 filename 是一个字符型变量内容为显示内容的文件的文件名。函数返回一个数组，数组长度为文件行数，文件的一行对应数组的一个元素。下面通过一个实例来说明 file()函数的使用方法。代码如下：

```
<html>
<head>
<title>file()函数使用实例</title>
</head>
<body>
<?
//使用 file 函数将整个文件内容读入到数组变量中
$filename="data.txt";                            //定义变量
$myfile=file($filename);                         //用 file()函数打开文件并赋值给变量
for($i=0;$i<count($myfile);$i++)                 //使用 for 循环
{
    echo $myfile[$i];                            //显示数组变量的每个元素
}
?>
</body>
</html>
```

保存以上代码为“8-10.php”。在 PHP 运行环境中执行以上代码，会发现其执行结果与图 8.2 一模一样。两个实例都是通过循环显示出了文件的所有内容。不同的是 fgetc()函数是逐个显示各个字符，而 file()则是把文件读入数组，再通过显示数组元素显示出文件内容的。

8.5　删除文件

如果确认文件已经没有用了，并且以后也不会再用到，就要把服务器上的文件删除以节省硬盘空间。在 PHP 中使用 unlink()函数来执行这一操作。

unlink(string filename) 函数，参数 filename 为字符变量，其内容为想要删除文件的文件名。执行后删除指定文件。如果出错，则返回 0 或者 False。当然，使用这一函数之前需要确认文件确实不再有用，执行 unlink()后将不可恢复。下面使用实例来介绍这一函数的使用方法。代码如下：

```
<html>
<head>
<title>unlink()函数使用实例</title>
</head>
<body>
<?
//使用 unlink 函数删除文件
$filename="data.txt";                          //定义变量
unlink($filename);                             //用 unlink 函数删除文件
?>
</body>
</html>
```

把以上代码保存为“8-11.php”。在 PHP 运行环境中执行以上代码，进入目录将会发现 data.txt 文件已经不见了，因为它已经被删除了。所以使用这一函数前，一定要确认文件已经不再需要，否则就再也无法恢复了。

使用此函数时需要注意的是，删除的文件必须是存在的文件。如果文件名并不存在，就会出现如图 8.4 所示的出错提示。



图 8.4　unlink()使用不当的出错提示

8.6　创建目录

有时需要在服务器上创建目录。如创建以当天日期为名字的目录来备份数据，或者创建以注册用户名为名字的目录来存放用户注册信息文件等。在 PHP 中使用 mkdir()函数来完成这一任务。

mkdir(string dirname,int mode)函数，参数 dirname 为字符变量，内容为想要创建目录的名称；参数 mode 为整型变量，表示创建模式。执行此函数将在指定目录下创建新的目录。下面通过一个例子来说明函数的使用方法。代码如下：

```
<html>
<head>
<title>mkdir()函数使用实例</title>
```

```
</head>
<body>
<?
//使用 mkdir 函数创建目录
$dirname="mydir";                                   //定义变量
mkdir($dirname,0700);                               //用 mkdir 来创建目录
?>
</body>
</html>
```

保存以上代码为“8-12.php”，并在 PHP 运行环境中执行以上代码，进入目录会发现多出了一个名字为 mydir 的目录，表示 mkdir 创建目录这句被顺利执行。

使用此函数时需要注意的是，创建的目录名不能与已经存在的目录名相同，如果出现了同样的目录名，就会出现如图 8.5 所示的出错提示。



图 8.5　mkdir()使用不当的出错提示

8.7　删 除 目 录

和文件一样，如果确认目录已经没有用，并且以后也不会再用到，就要把目录删除。在 PHP 中使用 rmdir()函数来执行删除目录的操作。

rmdir(string dirname)函数，参数 dirname 为字符变量，内容为想要删除目录的名称。执行此函数把指定目录删除。

下面通过一个例子来说明函数的使用方法。代码如下：

```
<html>
<head>
<title>rmdir()函数使用实例</title>
</head>
<body>
<?
//使用 rmdir 函数删除目录
$dirname="mydir";                             //定义变量
rmdir($dirname);                              //用 rmdir 来删除目录
?>
</body>
</html>
```

把以上代码保存为“8-13.php”。在 PHP 运行环境中执行以上代码，进入目录将会发现名为 mydir

的目录已经被删除，表示 rmdir 这句顺利被执行。

在使用这个函数时有两个注意事项：

- ❑ 目录必须为空，如果非空就会出现如图 8.6 所示的出错提示。
- ❑ 要删除的目录必须存在，如果删除不存在的目录，就会出现如图 8.7 所示的出错提示。



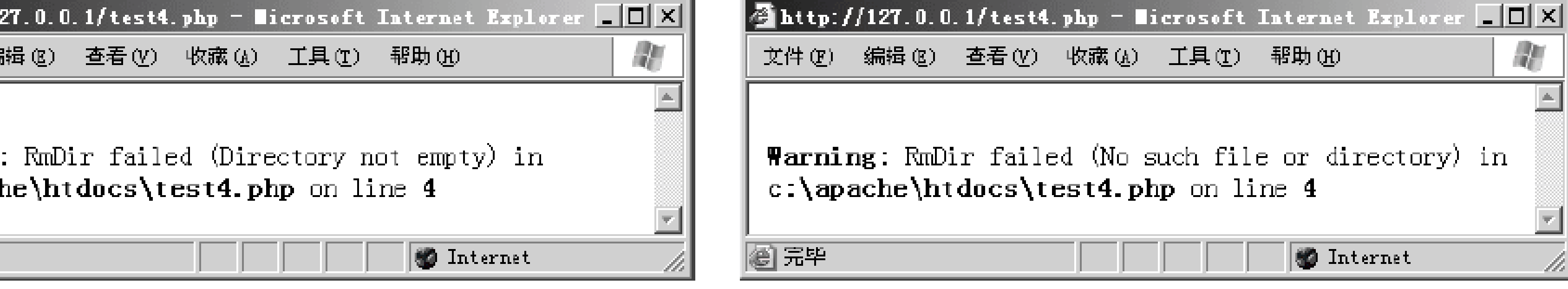


图 8.6　rmdir()使用不当的出错提示（1）　　图 8.7　rmdir()使用不当的出错提示（2）

8.8　浏览目录下的文件

在进行 PHP 编程的时候，需要对服务器某个目录下面的文件进行浏览，这时就需要用到 opendir()函数、readdir()函数和 closedir()函数，详见表 8.3。

表 8.3　查看目录下的文件相关函数

函　数　名	作　　用	参数及返回值
opendir(string path)	打开指定目录	参数path为目录的路径及目录名。函数返回值为可供其他目录函数使用的int型句柄
readdir(int dir_handle)	读取指定目录	参数dir_handle为已经用opendir()函数打开的可操作目录句柄。函数返回目录中的文件名称
closedir(int dir_handle)	关闭指定目录	参数dir_handle为已经用opendir()函数打开的可操作目录句柄。函数无返回值，运行后，将关闭打开指向dir_handle的目录

下面用一个实例来说明这 3 个函数的使用方法。注意，在使用该例子前请确保同一目录下有 data 文件夹。代码如下：

```
<html>
<head>
<title>浏览目录中的文件</title>
</head>
<body>
<table border="1">
<tr>
<td>文件名</td>
<td>文件大小</td>
<td>文件类型</td>
<td>修改时间</td>
</tr>
<?
$dirname="data";                                        //定义变量
```

```
$dir_handle=opendir($dirname);                                  //用 opendir 打开目录
while($file=readdir($dir_handle))                               //循环读取目录里的内容
{
    echo "<tr>";
    echo "<td>".$file."</td>";                                  //显示文件名
    echo "<td>".filesize($file)."</td>";                        //显示文件大小
    echo "<td>".filetype($file)."</td>";                        //显示文件类型
    echo "<td>".date("Y 年 n 月 t 日",filemtime($file))."</td>";   //格式化显示文件修改时间
    echo "</tr>";
}
closedir($dir_handle)                                           //关闭文件操作句柄
?>
</table>
</body>
</html>
```

把以上内容保存为一个名称为“8-14.php”文件，在 PHP 执行环境下运行这个文件，执行结果如图 8.8 所示。

浏览目录中的文件 - Microsoft Internet Explorer

文件(F)　编辑(E)　查看(V)　收藏(A)　工具(T)　帮助(H)

文件名	文件大小	文件类型	修改时间
.	0	dir	2006年8月31日
..	0	dir	2006年7月31日
index.html	1577	file	2002年2月28日
readme.txt	2221	file	2002年2月28日
data.txt	120	file	2006年8月31日
1.jpg	44087	file	2005年11月30日
2.jpg	53769	file	2005年11月30日
3.jpg	33581	file	2005年11月30日
4.jpg	44033	file	2005年11月30日
5.jpg	41998	file	2005年11月30日
test4.php.bak	514	file	2006年8月31日
test4.php	514	file	2006年8月31日

完毕　Internet

图 8.8　浏览目录中的文件

当然，显示细节会因为文件夹里内容的不同而有所不同。通过上例可见，在 PHP 中浏览文件夹中的内容也并不是一件多么复杂的事情。

8.9　关于文件上传

文件的上传也是 Web 应用程序文件操作中一个很重要的组成部分。因为大部分文件存在于用户客户端的机器上，如果要想让其他用户访问则必须要把它上传到服务器。如图片网站上传图片、软件下载站需要上传软件、音乐站点需要上传 MP3 音乐文件等。PHP 也提供了对文件上传的支持。本节就为读者介绍 PHP 中关于文件上传的操作。

PHP 是通过 Web 表单中的 FILE 组件来实现文件上传的。关于与浏览用户互动的 Web 表单将在本书第 15 章详细介绍，本节只是使用其中的 form 表单来实现文件上传的目的。

PHP 的文件上传主要是利用 form 表单提交一个 FILE 对象给服务器。其中 FILE 对象必须包含 Multipart/form-data 的 entype 属性。同时考虑到大文件上传容易造成网络超时的情况，所以可以用 set_time_limit($TimeLimit)来加大超时限制时间。

表单提交后，PHP 将检测上载的文件。文件存放在服务器上的一个临时目录中。同时生成一个与文件域（上传的文件）同名的几个变量，如_name 代表文件名称，_size 代表文件大小的字节数，_type 为文件的类型。由于临时目录会被删除，所以必须利用 copy()函数将文件复制到目标路径下。

下面给出文件上传的前台代码（因为前台只有 form 表单，并不涉及 PHP 操作，所以里面内容并没有 PHP 代码）。如下：

```
<html>
<head>
<title>文件上传的前台页面</title>
</head>
<body>
<form ENCTYPE="multipart/form-data" ACTION="8-16.php" METHOD="POST">
<input name="upfile" type="file">
<p>
<input type=submit value="确认提交">
<input type=reset value="重新选择">
</body>
</html>
```

把以上代码保存为“8-15.php”。先来解释一下，以上代码中的 form 表单有一个属性是 action="8-16.php"。因为这里要使用 8-16.php 来处理表单的提交，所以要把 form 表单的 action 属性指向该文件。

下面就来编写后台处理程序 8-16.php 的内容。代码如下：

```
<html>
<head>
<title>文件上传后台处理页面</title>
</head>
<body>
<?
if($upfile=="none")                                           //没有选定文件的处理
{
    echo "没有选择文件";                                        //显示提示信息
    echo "<p>";
    echo "点<a href=\"8-15.php\">这里</a>返回";                 //给出返回链接
}
else                                                          //选定文件
{
    $filepath="images/";                                      //定义路径
    $filename=$filepath.$upfile_name;                         //新的路径及文件名
    echo $upfile_name;                                        //显示文件名
    echo "<p>";
    echo $upfile_size;                                        //显示文件大小
    echo "<p>";
```

```
        echo $upfile_type;                                      //显示文件类型
        if(copy($upfile,$filename))                              //复制文件的目标路径
        {
            unlink($upfile);                                    //删除原有文件
            echo "<p>";
            echo "指定文件已经成功上传！";
            echo "<p>";
            echo "点<a href=\"8-15.php\">这里</a>返回";          //给出返回链接
        }
        else
        {
            echo "文件上传失败!";
        }
    }
    ?>
    </body>
    </html>
```

保存以上代码为“8-16.php”文件。在 PHP 执行环境中运行 8-15.php 文件，其执行结果将会如图 8.9 所示。然后，先不选择任何文件，而是直接单击“确认提交”按钮，提交表单，将会出现如图 8.10 所示的执行结果。



图 8.9　文件上传前台页面执行结果



图 8.10　不选择任何文件时的后台处理结果

单击图 8.10 中的“这里”链接，返回 8-15.php。这时选择一个文件，如“1.jpg”文件，然后单击“确认提交”按钮(在执行这一步之前，请确认 8-15.php 的同一目录下有 images 这样一个目录)，8-16.php 将会返回如图 8.11 所示的结果。

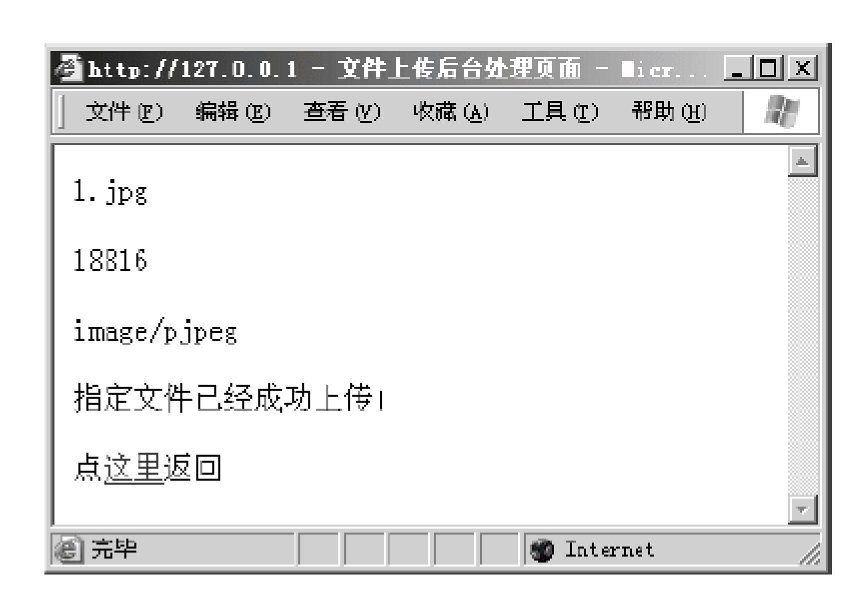


图 8.11　正确选择文件上传的执行结果

从如图 8.11 所示的结果可以看到，程序正确执行了文件上传命令。打开 8-16.php 同一目录下的 images 文件夹，发现里面多了一个名为 1.jpg 的图片文件，说明整个上传过程顺利完成。

8.10　文件操作综合实例：在线相册

本节将综合本章所学习的知识，来建立一个简单的基于文件及文件上传的在线相册系统。在线相册拥有图片文件上传、图片浏览和图片删除等功能。通过本节的学习，将会把本章所学的知识串联起来，使读者进一步熟悉 PHP 中关于文件的操作。

8.10.1　系统功能

整个相册管理系统要实现的功能有图片文件的上传操作；图片文件的浏览操作和图片文件的删除操作等。下面分几小节来解释创建构成本系统的几个重要文件。这里先列出代码，然后再对每一个文件进行解释。

8.10.2　相册系统首页面

8-17.php 为该系统的第一个显示页面，它将读取存放数据的文本文件的内容，从而显示出相应的图片。并且其中还使用了分页显示，如果上传的总图片数多于 8 幅，则会显示下一页的链接。然后，判断所在的页，并显示出相应的内容。代码如下：

```
<html>
<head>
<title>简易相册系统首页</title>
</head>
<body>
<center>
<h1>简易相册系统首页</h1>
<p>
<a href="8-18.php">上传图片</a><p>
<?
if(!$_GET["page"])                                    //如果没有参数 page
$page=1;                                              //则显示第一页内容
else
$page=$_GET["page"];                                  //如果带有参数 page，则显示相应页内容
$filename="data.dat";                                 //指定记录数据文件名
$myfile=file($filename);                              //使用 file()函数把文件所有信息读入一个数组
$z=$myfile[0];                                        //把数组第一条内容赋值给变量
if($z=="")                                            //如果文件为空，即没有任何图片上传
echo "目前记录条数为：0";                              //显示没有记录的信息
else                                                  //如果有图片上传
{
      $temp=explode("||",$myfile[0]);                 //读出数组第一条记录到数组
      echo "共有".$temp[0]."条内容";                   //读出该数组第一个元素（代表记录总条数）
      echo "    ";
```

```
        $p_count=ceil($temp[0]/8);                    //计算总页数为记录总条数除以每页显示条数
        echo "分".$p_count."页显示";                   //输入总页数
        echo "    ";
        echo "当前显示第".$page."页";                   //当前页
        echo "<table border='1'>";
        if($page!=ceil($temp[0]/8))                   //如果当前页不是最后一页
        $current_size=8;                              //当前页最多可显示 8 条记录
        else                                          //如果当前页是最后一页
        $current_size=$temp[0]%8;                     //当前页显示的条数为总条数除以 8 的余数
        if($current_size==0) $current_size=8;         //如果正好是 8 的倍数则显示 8 条内容
        for($i=0;$i<ceil($current_size/4);$i++)       //通过循环输出行，每行四列
        {
            echo "<tr>";
            for($j=0;$j<4;$j++)                       //通过循环输出单元格，共 4 个
            {
                echo "<td>";
                $temp=explode("||",$myfile[$i*4+$j+($page-1)*8]);//把相应的记录按“||”分割到数组
                if(($i*4+$j+($page-1)*8)<$z)          //如果当前数小于总数显示图片
                {
                    $imgfile="images\\".$temp[1];     //显示图片为数组的第 2 个元素
                    $flag=getimagesize($imgfile);     //获得图片的大小以加以大小处理
                    echo "<a href=8-20.php?id=".$temp[0]."><img src=images\\".$temp[1];
                    if($flag[0]>180||$flag[1]>100)    //如果图片太大
                    echo " width=180 height=".$flag[1]*180/$flag[0];
                    echo " border=\"0\"></a>";        //把图片按比例缩放显示
                }
                else                                  //如果当前数比总记录数大
                echo "暂时没有图片";                    //输出没有图片的信息
                echo "</td>";                         //结束该单元格
            }
            echo "</tr>";                             //结束行
        }
        echo "</table>";                              //结束表格
    }
    echo "<p>";
    //以下内容为分页显示链接
    $prev_page=$page-1;                               //前一页
    $next_page=$page+1;                               //下一页
    if ($page<=1)                                     //如果当前页小于等于 1
    {
        echo "第一页 | ";
    }
    else                                              //如果当前页大于 1
    {
        echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
    }
    if ($prev_page<1)                                 //如果前一页小于 1
    {
        echo "上一页 | ";
```

```
}
else                                                    //如果前一页大于等于 1
{
    echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
}
if ($next_page>$p_count)                                //如果下一页大于总页数
{
    echo "下一页 | ";
}
else                                                    //如果下一页小于等于总页数
{
    echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
}
if ($page>=$p_count)                                    //如果当前页大于等于总页数
{
    echo "最后一页</p>\n";
}
else                                                    //如果当前页小于总页数
{
    echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
}
?>
</center>
</body>
</html>
```

保存以为代码为“8-17.php”，以备后用。

8.10.3　相册系统上传前台页面

接下来制作上传图片的前台页面 8-18.php。这一页面中没有太多的 PHP 技术，不过利用了 JavaScript 技术，来实现对内容是否为空的判断，以及预览将要上传的图片。关于 JavaScript 的内容，这里不进行介绍，请读者参阅相关书籍。代码如下：

```
<html>
<head>
<title>相册管理系统图片上传页面</title>
</head>
<body>
<script language="javascript">
function Juge(theForm)
{
    if (theForm.upfile.value == "")
    {
        alert("请先选择文件！");
        theForm.upfile.focus();
        return (false);
    }
```

```
        if (theForm.content.value == "")
        {
                alert("请输入图片说明！");
                theForm.content.focus();
                return (false);
        }
        if (theForm.content.value.length>60)
        {
                alert("图片说明内容太多了，请删除一点再发！");
                theForm.content.focus();
                return (false);
        }
}
function viewmypic(img1,upfile)
{
        if (upfile.value)
        {
                img1.src=upfile.value;
                img1.style.display="";
                img1.border=1;
        }
}
</script>
<center>
<h1>相册管理系统图片上传页面</h1>
<p>
<a href="8-17.php">返回首页</a>
<table border="1">
<form ENCTYPE="multipart/form-data" ACTION="8-19.php" METHOD="POST" onsubmit="return Juge(this)">
<tr>
<td>选择图片：</td>
<td><input name="upfile" type="file" onchange=viewmypic(img1,this.form.upfile)></td>
</tr>
<tr>
<td>输入说明：</td>
<td><input name="content" type="text">(*限 30 字)</td>
</tr>
<tr>
<td colspan="2"><center><input type=submit value="确认提交">
<input type=reset value="重新选择"></center></td>
</tr>
<tr>
<td colspan="2"><center><img src="" id="img1" style="display:none;"></center></td>
</tr>
</table>
</center>
</body>
</html>
```

把以上代码保存为“8-18.php”，以备后用。其中的 form 表单的 enctype 属性必须为 multipart/form-data，这是实现文件上传的关键。另外，表单的 action 属性指向 8-19.php，这就是处理上传结果的页面。8.10.4 小节就来创建这个页面。

8.10.4　相册系统上传后台页面

该页面实现的功能不仅要把文件上传到目标文件夹，而且还要把相关信息写入到记录文件之中。所以这里用到较多的 PHP 文件操作的内容。代码如下：

```
<html>
<head>
<html>
<head>
<title>相册管理系统后台处理页面</title>
</head>
<body>
<?
if($upfile=="none")                                          //如果没有选择相应的文件
{
    echo "没有选择文件";                                     //输出信息
    echo "<p>";
    echo "点<a href=\"8-18.php\">这里</a>返回";              //给出返回链接
}
else
{
    $filepath="images/";                                     //定义上传的路径
    $filename=$filepath.$_FILES[upfile][name];;                  //定义文件名
    if(copy($upfile,$filename))                              //如果文件被顺利复制
    {
        unlink($_FILES[upfile][tmp_name]);                   //删除临时文件
        $dataname="data.dat";                                //定义记录文件名
        $myfile=file($dataname);                             //使用 file()把记录文件按行读入数组
        if($myfile[0]=="")                                   //如果记录文件为空
        {
            $fp=fopen($dataname,"a+");
            fwrite($fp,"1||".$_FILES[upfile][name]."||".$_POST["content"]."||".date(Y 年 m 月 d 日)."\n");
            fclose($fp);                                     //直接写入行号为 1 的内容
        }
        else                                                 //如果记录文件非空即已经有内容
        {
            $temp=explode("||",$myfile[0]);                  //把第一条记录按“||”分割到数组
            $temp[0]++;                                      //得出总记录数并自增 1
            $fp=fopen($dataname,"r");                        //以只读方式打开文件
            $line_has=fread($fp,filesize("$dataname"));      //使用 fread 读出文件已经存在的内容
            fclose($fp);                                     //关闭文件
            $fp=fopen($dataname,"w");                        //以写入方式打开文件
            fwrite($fp,$temp[0]."||".$_FILES[upfile][name]."||".$_POST["content"]."||".date(Y 年 m 月 d 日)."\n");
```

```
                                                              //写入新的内容
            fwrite($fp,"$line_has");                          //写入原来已经存在的内容
            fclose($fp);                                      //关闭文件
        }
        echo "<p>";
        echo "指定文件已经成功上传！";
        echo "<p>";
        echo "点<a href=\"8-17.php\">这里</a>返回";            //给出返回链接
    }
    else
    {
        echo "文件上传失败!";                                  //如果没有复制相应文件显示失败信息
    }
}
?>
</body>
</html>
```

把以上代码保存为“8-19.php”，以备后用。现在文件上传的任务完成了，8.10.5 小节将创建显示图片文件全部信息页面。

8.10.5　相册系统浏览图片详细信息页面

通过 8.10.3，8.10.4 小节的页面，文件上传任务已经完成了。不过还不够完善，因为现在系统还只限于上传图片。如何浏览图片文件的全部信息呢？这一小节来创建页面用于浏览文件的详细信息。其实就是利用 explode()函数把记录内容分割到数组中，再分别显示数组各个元素。具体内容参看以下代码：

```
<html>
<head>
<title>简易相册系统查看图片</title>
</head>
<body>
<center>
<h1>简易相册系统查看图片页</h1>
<p>
<?
if(!$_GET["id"])                                          //如果没有指定 ID
{
    echo "没有指定 ID";                                    //输出相应信息
    echo "<p>";
    echo "点<a href=\"8-17.php\">这里</a>返回";            //给出返回链接
    exit();
}
else                                                      //如果有 ID
{
?>
<a href="8-17.php">返回首页</a>    
```

```
<a href="8-21.php?id=<? echo $id ?>">删除图片</a><p>
<?
    $id=$_GET["id"];                                    //把参数赋值给变量
    $filename="data.dat";                               //定义记录文件
    $myfile=file($filename);                            //使用 file()函数把文件按行读入数组
    $z=$myfile[0];                                      //把数组第一个变量赋值为变量
    if($z=="")                                          //如果记录数为 0
    echo "目前记录条数为：0";                           //显示相应内容
    else                                                //如果有内容
    {
        $temp=explode("||",$myfile[$z-$id]);            //用 explode()函数按“||”把相应记录分割
        echo "<table border='1'>";
        echo "<tr>";
        echo "<td>";
        echo "文件名：".$temp[1];                       //显示数组第二个元素即文件名
        echo "</td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td>";
        echo "<img src=\"images\\".$temp[1]."\">";      //显示图片
        echo "</td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td>";
        echo "图片简介：".$temp[2];                     //显示图片第三个元素即图片简介
        echo "</td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td>";
        echo "上传日期：".$temp[3];                     //显示图片第四个元素即上传日期
        echo "</td>";
        echo "</tr>";
        echo "</table>";
    }
}
?>
</center>
</body>
</html>
```

把以上代码保存为“8-20.php”，以备后用。

一般情况下，有了上述几个文件，已经可以满足基本需要了。但是，如何把没有价值的图片从服务器上删除呢？所以还需要一个页面用于实现图片的删除操作。8.10.6 小节就来创建实现图片删除操作的文件。

8.10.6　相册系统图片删除页面

要删除文件其实很简单，使用 unlink()函数就可以了。可现在问题是不光要把文件删除，而且还要

对记录文件作相应的修改。删除文件前的内容保持不变，删除后的每条记录还得-1。下面就给出能实现该功能的文件。具体内容参看以下代码：

```
<html>
<head>
<title>相册管理系统删除图片处理页面</title>
</head>
<body>
<?
if(!$_GET["id"])                                    //如果没有指定 ID
{
    echo "没有指定 ID";                              //显示相应信息
    echo "<p>";
    echo "点<a href=\"8-17.php\">这里</a>返回";      //给出返回链接
    exit();
}
else                                                //如果有 ID
{
    $id=$_GET["id"];                                //把参数 ID 赋值给变量
    $filename="data.dat";                           //定义记录文件
    $myfile=file($filename);                        //使用 file()函数把文件按行读入到数组
    $z=$myfile[0];                                  //数组第一个元素赋值给变量
    if($z=="")                                      //如果第一行为空
    echo "目前记录条数为：0";                        //输出没有记录
    else                                            //如果记录非空
    {
        $temp=explode("||",$myfile[$z-$id]);        //使用 explode 分割相应记录到数组
        $filepath="images/";                        //定义路径
        $imgfile=$filepath.$temp[1];                //获得文件名
        unlink($imgfile);                           //删除文件
        for($i=0;$i<($z-$id);$i++)                  //从第一条记录读到欲删除的记录
        {
            $temp2=explode("||",$myfile[$i]);       //使用 explode 分割相应记录到数组
            $temp2[0]--;                            //记录号实现自减
            $text2=$text2.$temp2[0]."||".$temp2[1]."||".$temp2[2]."||".$temp2[3];
                                                    //把新的内容赋值到变量
        }
        for($i=($z-$id+1);$i<$z;$i++)               //新欲删除的后一条记录到最后一条记录
        {
            $text1=$text1.$myfile[$i];              //内容保持不变
        }
        $fp=fopen($filename,"w");                   //以写入方式打开文件（文件同时被清空）
        fwrite($fp,$text2);                         //写入欲删除记录之前自减后的所有记录
        fwrite($fp,$text1);                         //写入欲删除记录后的所有记录
        fclose($fp);                                //关闭文件
        echo "指定文件已经删除成功！";
        echo "<p>";
        echo "点<a href=\"8-17.php\">这里</a>返回";
    }
}
```

```
?>
</body>
</html>
```

把以上代码保存为“8-21.php”，以备后用。

到这里，整个相册管理系统就基本上完成了。既能实现文件上传、浏览，又可以实现对图片文件的删除操作。下面，来看一看该系统的执行结果，更直观地感受一下编程带来的乐趣。

8.10.7　测试相册系统

在运行此系统之前要做一些简单的准备工作。第 1，在系统同一目录下建立一个名为 images 的文件夹；第 2，在系统同一目录下建立一个名为 data.dat 的文本文件。读者可以先建一个文本文件，里面什么也不要输入。然后把扩展名改为 dat 即可。

（1）运行 8-17.php。第 1 次运行，执行结果如图 8.12 所示。由于此时还没有任何图片被上传，记录文件里的记录是空的。

（2）单击“上传图片”链接，进行图片的上传。单击链接后，出现如图 8.13 所示的执行结果。

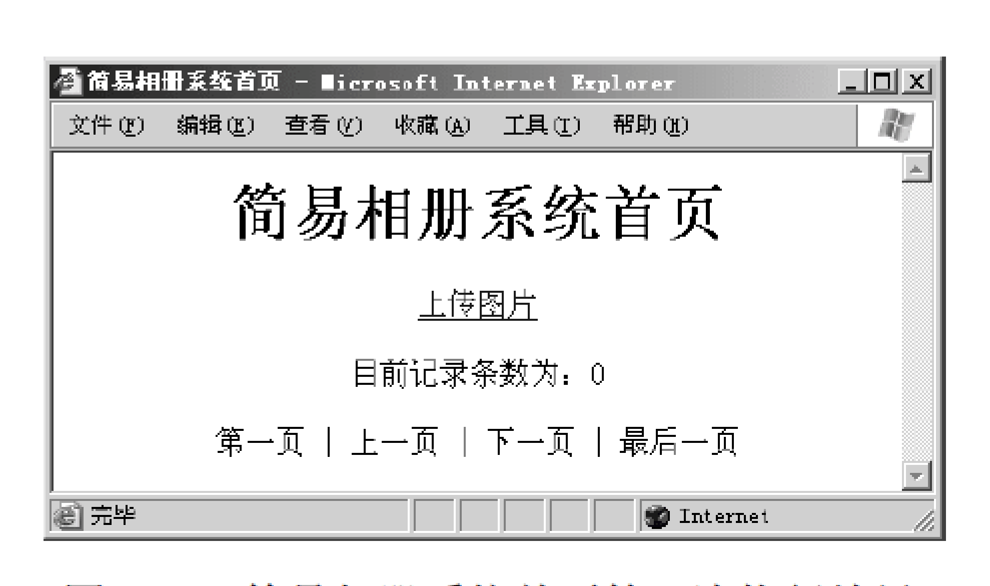


图 8.12　简易相册系统首页第 1 次执行结果



图 8.13　相册管理系统图片上传页面执行结果

（3）单击“浏览”按钮，选择相应的图片。页面最下方出现相应图片的预览效果，如图 8.14 所示。

（4）输入相应的图片说明，此处输入“用户图像第 1 幅图”，单击“确认提交”按钮，就可把图片上传到服务器上了。

由于 8-19.php 的输出结果很简单，只是简单的提示上传成功，然后给出一个返回首页的链接。上传了一幅图片的 8-17.php 的效果如图 8.15 所示。

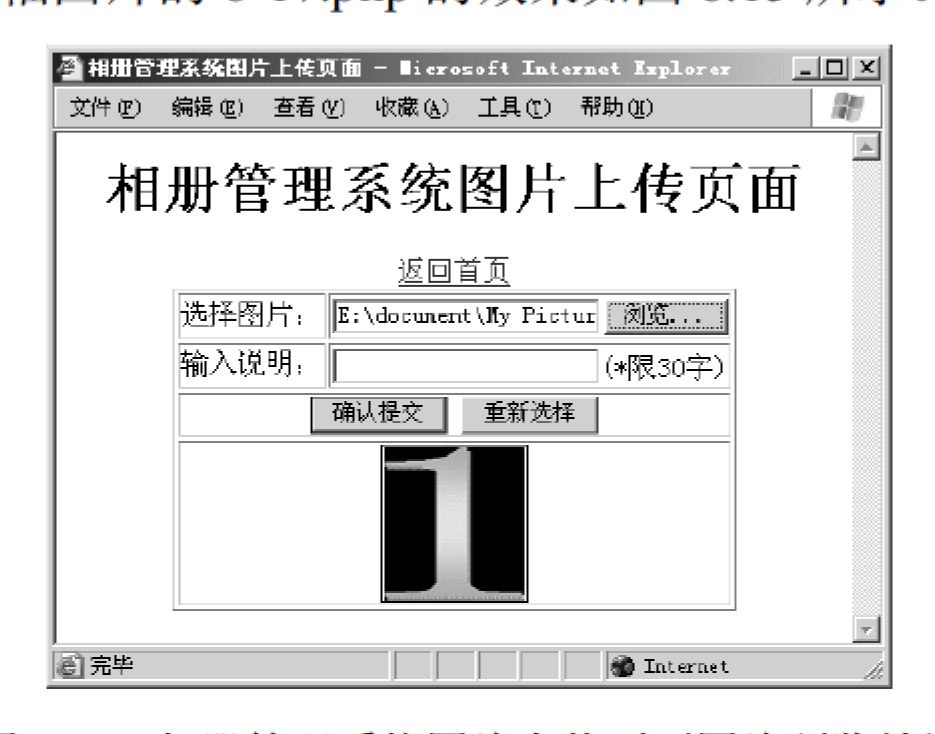


图 8.14　相册管理系统图片上传页面图片预览效果



图 8.15　上传一幅图片后的简易相册系统首页执行结果

打开目录下的 images 文件夹，查看里面也增加了一个图片文件。再查看相册系统目录下的 data.dat 文件，里面显示如下信息。

```
1||1.gif||用户图像第 1 幅图||2006 年 09 月 20 日
```

这是图片的详细信息，包括图片的序号、文件名、图片说明和上传日期等。说明记录文件正确记录了相关内容。此时可以直接单击图片，以查看图片详细信息。打开后的效果如图 8.16 所示。

这里显示出了已经上传图片的详细信息。单击“返回首页”链接，以继续上传图片。

现在继续上传图片，以验证系统的分页显示功能是否正常运行。如果上传的图片超过了 8 幅，8-17.php 的下方就会显示出到另外页的链接，如图 8.17 所示。







图 8.16　简易相册系统查看图片详细信息页　　图 8.17　简易相册系统首页的分页功能起了作用

可以看出，最上面显示的内容是“共有 9 条内容，分 2 页显示，当前显示第 1 页”，而最下方则显示了“下一页”及“最后一页”的链接。单击“下一页”链接，执行结果如图 8.18 所示。

这里显示的是第 2 页的内容，说明分页功能正常。这时再单击“上一页”链接，就回到了第一页。

这时，打开 data.dat 文件，里面将会有如下内容：

```
9||9.gif||用户图像第 9 幅图||2006 年 09 月 20 日
8||8.gif||用户图像第 8 幅图||2006 年 09 月 20 日
7||7.gif||用户图像第 7 幅图||2006 年 09 月 20 日
6||6.gif||用户图像第 6 幅图||2006 年 09 月 20 日
5||5.gif||用户图像第 5 幅图||2006 年 09 月 20 日
4||4.gif||用户图像第 4 幅图||2006 年 09 月 20 日
3||3.gif||用户图像第 3 幅图||2006 年 09 月 20 日
2||2.gif||用户图像第 2 幅图||2006 年 09 月 20 日
1||1.gif||用户图像第 1 幅图||2006 年 09 月 20 日
```

记下这些内容，以和删除图片的内容作比较。接下来，来验证一下删除用户上传图片功能是否能正确运行。操作步骤如下：

（1）单击如图 8-16 所示的图中第 1 行第 4 个图形，将出现查看图片详细信息的页面，如图 8.19 所示。







图 8.18　简易相册系统首页的分页功能测试　　　图 8.19　删除图片前的准备

由于系统没有设定对删除进行确认的功能，所以一旦单击了“删除图片”链接，对应的图片就会被删除。这时要记得已经上传了 9 幅图片，分两页显示。

（2）单击“删除图片”链接，直接执行 8-21.php。执行完毕会提示删除成功，直接给出返回首页的链接。回到首页，结果如图 8.20 所示。



图 8.20　删除了一幅图片后的结果

内容又成了 8 条，并且那一幅图片 6.gif 已经不见了。这时打开 data.dat 文件，里面内容如下：

```
8||9.gif||用户图像第 9 幅图||2006 年 09 月 20 日
7||8.gif||用户图像第 8 幅图||2006 年 09 月 20 日
6||7.gif||用户图像第 7 幅图||2006 年 09 月 20 日
5||5.gif||用户图像第 5 幅图||2006 年 09 月 20 日
4||4.gif||用户图像第 4 幅图||2006 年 09 月 20 日
3||3.gif||用户图像第 3 幅图||2006 年 09 月 20 日
2||2.gif||用户图像第 2 幅图||2006 年 09 月 20 日
1||1.gif||用户图像第 1 幅图||2006 年 09 月 20 日
```

与前面的 data.dat 文件内容相比较，后面的都没变，前面的序号都减少了 1，说明正常完成了删除图片的操作。

至此，本系统测试完毕。通过对这个图片管理系统的调试及运行可以看到，里面包含了大量的文

件操作的内容及技巧，对本章所学的内容是一个很好的检阅。读者一定要认真领会其中的内容，学以致用，把学到的内容用到实际的工作中去。

但是，这个系统还有一些需要完善的地方。这里简要介绍一下，请读者自己完成。

- 并没有对用户上传内容作判断。用户可能会上传不是图片的文件，这是应该要判断的。解决办法就是判断文件的扩展名，或者使用 getimagesize()函数来判断图片类型，如果不能正确返回值，就证明不是图片，不予以上传。
- 也是前面提到的问题，没有对用户删除图片进行确认。如果用户误单击了链接就不可恢复了。解决办法是再建一个中间页面，或者在 8-21.php 中带上一个参数。如果没有参数则只是给出警告信息，其中有指向本页带参数的链接，如果用户仍然单击了链接，那么就执行删除文件操作。或者把链接做如下简单的改动，可以调用 JS 的 confirm 方法来达到确认的目的。

以下是未改动以前的链接：

```
<a href="8-21.php?id=<? echo $id ?>">删除图片</a><p>
```

改为如下样式：

```
<a onclick="c()" href="8-21.php?id=<? echo $id ?>">删除图片</a>
```

在链接中加入了一个 onclick 事件，然后再在 8-20 的源文件中加入以下 JavaScript 代码即可。

```
<script language=javascript>
function c()
{
    temp=confirm("确定要删除图片吗？");
    if(temp==0) return else;
}
</script>
```

- 系统存在安全隐患。因为只要是浏览该页面的用户均可以上传图片，同时也能删除图片。解决办法是判断用户来源，或者直接给上传及删除页面加上密码，只有密码正确才能执行相关操作。这样就提高了系统的安全性。

8.11　本 章 小 结

本章主要介绍了 PHP 编程中对文件的操作。包括判断文件存在、获取文件属性、读取文件内容、增删目录、浏览目录等文件操作，同时还讲解了文件上传的操作，学习了 PHP 文件类库函数的使用与技巧。因为文件的操作在 PHP 编程中有着广泛的应用，所以熟练掌握这些函数的使用及文件操作技巧，对 PHP 的 Web 编程将起到事半功倍的效果。

第 9 章

用 PHP 获取主机信息

- 使用 phpinfo()获取主机信息
- 获取用户相关信息
- 使用 date()获取主机日期信息
- 获取特定日期的信息

在使用 PHP 进行网络编程的时候，让用户了解主机的信息及使用者的情况是十分有必要的。例如，获取主机的操作系统类型、Apache 的版本、PHP 解释器的版本、有无 MySQL 数据库支持以及主机的日期、时间等内容。这一章来介绍 PHP 中常用的主机函数，包括 phpinfo()函数的使用、获取浏览器相关信息、使用日期时间函数获取相应内容等。通过本章的学习，读者对于如何获得主机信息将有一个深层次的了解。

9.1　phpinfo()的使用

如果用户需要全面了解服务器的相关信息，使用 phpinfo()函数是最为方便的。该函数没有参数，直接返回主机的全部信息，使用非常方便，返回内容相当广泛。正因为该函数包括了太多的内容，甚至一些不希望被用户发现的内容，所以许多服务提供商往往禁用了该函数。

下面通过一个实例来说明如何使用 phpinfo()函数及它的返回内容。具体内容参看以下代码：

```
<html>
<head>
<title>phpinfo()函数使用实例</title>
</head>
<body>
<?
phpinfo();                                        //显示相关信息
?>
</body>
</html>
```

以上代码相当简单，起实质作用的只有“phpinfo()”这一句 PHP 代码。保存以上代码为“9-1.php”然后在 PHP 执行环境中运行以上代码，其执行结果如图 9.1 所示。

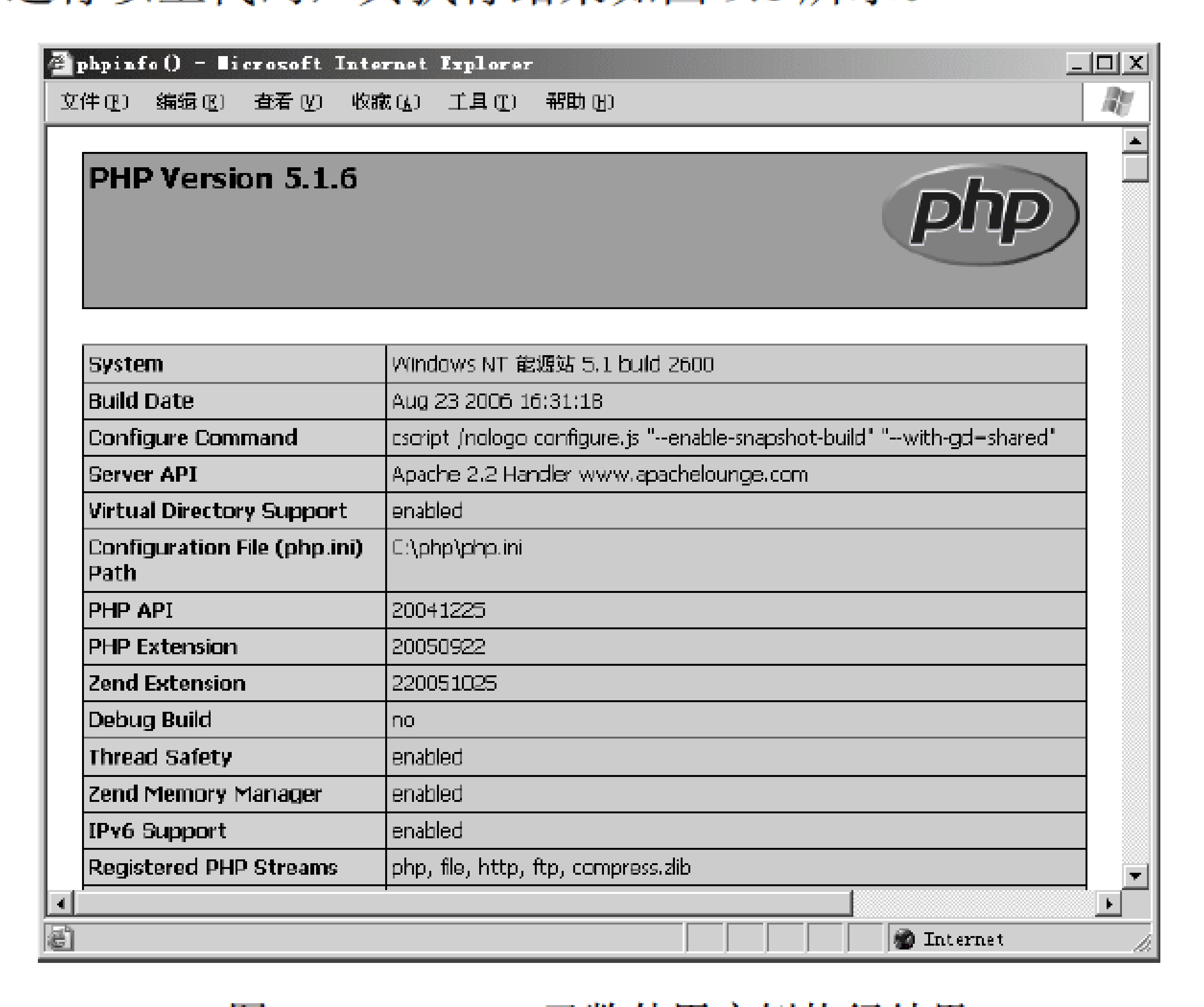


图 9.1　phpinfo()函数使用实例执行结果

通过图 9.1 可见，phpinfo()函数返回的内容相当多，几乎包括了所有与 PHP 主机有关的内容。

9.2　浏览器相关操作

在使用 PHP 编写 Web 应用程序时，获取用户浏览器相关信息也是十分必要的。如使用者所使用的

浏览器信息、操作系统的类型、访问者客户机的 IP 地址等。本节就来介绍 PHP 中的浏览器相关操作的内容。

9.2.1　获取访问者浏览器信息

流行的浏览器有很多种，如微软公司的 IE、网景公司的 Netscape、Opera 以及火狐（Firefox）、遨游等。不同的浏览器支持的标准不同，对相同的 HTML 也有不同的解释，所以判断浏览器的类型从而为用户提供相应的网页是很有必要的。在 PHP 中判断浏览器的类型也是一件很简单的事情，简单到只使用系统的内置系统变量$_SERVER['HTTP_USER_AGENT']即可。

在第 3 章介绍到的系统变量中曾经提到过$_SERVER 是一个特殊的 PHP 保留变量，它包含了主机提供的所有信息，被称为自动全局变量（或“超级全局变量”）。要显示该变量，只需简单地进行如下操作。

```
<?
echo $_SERVER["HTTP_USER_AGENT"];
?>
```

该脚本的输出可能是 Mozilla/4.0(compatible; MSIE 5.01; Windows NT5.0)。

下面通过一个实例来说明在 PHP 中如何判断用户浏览器类型。

```
<html>
<head>
<title>判断浏览器类型使用实例</title>
</head>
<body>
<?
echo "您的浏览器类型为：";
echo "<p>";
if(strstr($_SERVER["HTTP_USER_AGENT"], "MSIE"))      //使用系统变量判断用户浏览器类型
{
     echo"Internet Explorer";
}
?>
</body>
</html>
```

保存以上代码为“9-2.php”。在 PHP 执行环境中运行以上代码，其执行结果如图 9.2 所示。



图 9.2　判断浏览器类型使用实例执行结果

9.2.2　获取访问者的 IP 地址

在实际应用中，判断用户 IP 也是很有必要的。如网上投票程序就要用 IP 来判断用户是否已经投过票、限制 IP 计数器用 IP 地址来判断当天用户是否已经浏览过站点。在 PHP 中的$_SERVER 系统变量包含了主机的几乎所有信息，所以 IP 地址自然也能用这个变量获得，其获得方法是使用$_SERVER[REMOTE_ADDR]。

下面通过一个实例来说明如何使用$_SERVER 系统变量来获得访问者的 IP 地址。具体内容参看以下代码：

```
<html>
<head>
<title>获得访问者 IP 地址使用实例</title>
</head>
<body>
<?
echo "您的 IP 地址为：";
echo "<p>";
echo $_SERVER[REMOTE_ADDR];              //使用系统变量输出用户 IP
?>
</body>
</html>
```

保存以上代码为“9-3.php”。在 PHP 执行环境中运行以上代码，执行结果如图 9.3 所示。

从图 9.3 中可以发现正确显示了本机的 IP（由于是在单机上进行调试程序，所以返回的只是单机的 IP 地址）。

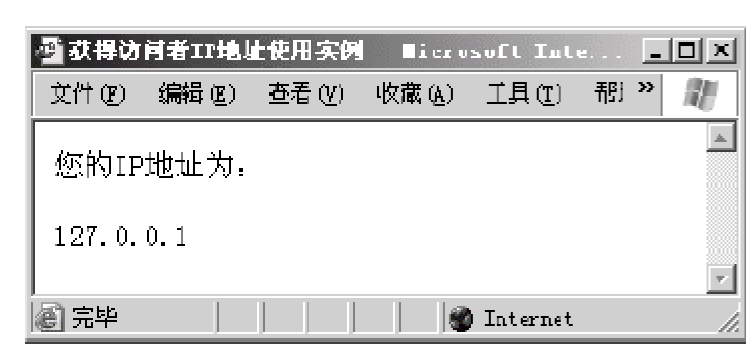


图 9.3　获得访问者 IP 地址使用实例执行结果

9.3　日期时间相关函数

在编程中了解系统的日期及时间也是很有必要的。如在制作留言簿时，需要记录用户的留言日期及时间；论坛中新用户的注册日期及时间等。在 PHP 中要获得系统的日期及时间是很容易的，这一节就为读者分别作介绍。

9.3.1　检查日期的合法性

用户的输入内容不一定是想要取得的内容。如让用户输入生日，用户很有可能输入“2 月 30 日”这样错误的日期，而错误的日期入库，将会给统计及以后的操作带来很多不便。所以判断用户输入的日期是否正确是很有必要的。

PHP 中有专门用来检查日期是否合法的函数 checkdate()。如果日期合法就返回 True，反之则返回

False。下面通过一个实例来说明 checkdate()函数是如何使用的。

```
<html>
<head>
<title>检查日期合法性函数使用实例</title>
</head>
<body>
<?
if(checkdate(12,30,2005))                        //如果检查日期合法
echo "12,30,2005 是正确的日期！";
else                                             //如果检查日期不合法
echo "12,30,2005 不是正确是日期！";
echo "<p>";
if(checkdate(2,30,2005))                         //如果检查日期合法
echo "2,30,2005 是正确的日期！";
else                                             //如果检查日期不合法
echo "2,30,2005 不是正确是日期！";
?>
</body>
</html>
```

保存以上代码为“9-4.php”，在 PHP 执行环境中运行以上代码，执行结果将会如图 9.4 所示。

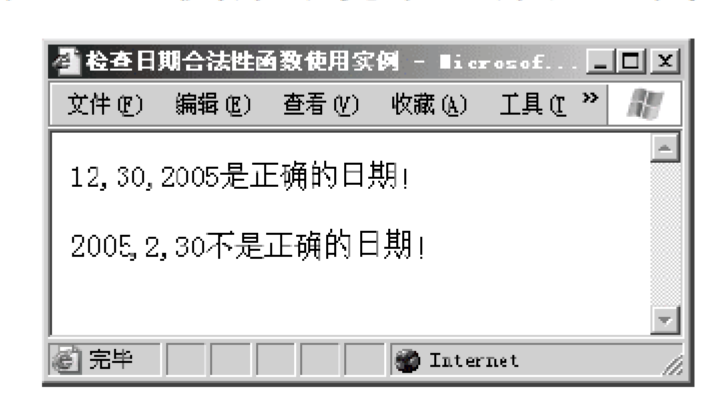


图 9.4　检查日期合法性函数使用实例执行结果

学习过英文的读者都知道西方的日期表示习惯是“月、日、年”。所以从图 9.4 结果来看，12 月 30 日 2005 年是一个正确的日期；而 2005 年 2 月 30 日则是一个错误的日期。9-4.php 通过 checkdate()函数判断日期的正确性，然后判断返回值并打印相应的信息。

9.3.2　格式化输出当前日期

这一小节来介绍 PHP 中如何获得当前日期。在 PHP 中使用 date()函数加上不同的参数可以获得不同的日期时间信息。具体内容如表 9.1 所示。

表 9.1　date()函数的参数意义及返回值

参　　数	表 示 意 义	返　回　值
d	表示显示月份中的第几天	返回有前导零的2位数字，从01到31
D	表示星期中的第几天，以文本形式显示	返回3个字母从Mon到Sun
j	表示月份中的第几天	返回没有前导零的数字，从1到31
l（“L”的小写字母）	表示星期几	返回完整的文本格式从Sunday到Saturday

续表

参　数	表 示 意 义	返　回　值
S	表示每月天数后面的英文后缀	返回2个字符，如：st、nd、rd或者th。可以和j一起用
w	表示星期中的第几天	返回0（星期天）到6（星期六）的数字
z	表示年份中的第几天	返回从0到366到数字
F	表示月份	返回完整的文本格式，如：January或者March，从January到December
m	以数字形式表示的月份	返回有前导零的2位数字，从01到12
M	以字母形式表示的月份	返回3个字母缩写表示的月份从Jan到Dec
n	以数字形式表示的月份（与m不同）	返回无前导零的位数字，从1到12
t	表示一月的天数	返回给定月份所应有的天数，从28到31
L	表示是否为闰年	如果是闰年返回1，否则返回0
Y	表示年份	返回4位数字完整表示的年份，如：1999或2003
y	表示年份（与Y不同）	返回2位数字表示的年份，如：99或03
a	以小写字母表示上午或下午	返回值为am或pm
A	以大写字母表示上午或下午	返回值为AM或PM
g	表示小时，12小时格式	返回没有前导零的表示小时的数字，从1到12
G	表示小时，24小时格式	返回没有前导零的表示小时的数字，从0到23
h	表示小时，12小时格式（与g不同）	返回有前导零的表示小时的两位数字，从01到12
H	表示小时，24小时格式（与G不同）	返回有前导零的表示小时的两位数字，从00到23
i	表示分钟	返回有前导零的分钟数，从00到59
s	表示秒	返回有前导零的秒数，从00到59
T	表示本机所在的时区	如：EST，MDT等。在Windows下为完整文本格式，如“EasternStandardTime”。中文版会显示“中国标准时间”

下面用一个实例来说明如何获得系统主机的日期及时间，具体内容参看以下代码。

```
<html>
<head>
<title>输出当前日期函数使用实例</title>
</head>
<body>
<?
echo "今天的日期是：";
echo "<p>";
echo date(Y 年 n 月 d 日);                              //格式化日期
echo "<p>";
echo "今天是星期：";
echo date(w);                                           //输出星期
echo "<p>";
echo "现在的时间是：";
echo "<p>";
```

```
echo date(aG 点 i 分 s 秒);
?>
</body>
</html>
```

保存以上代码为“9-5.php”。然后在 PHP 执行环境中运行该文件，其执行结果如图 9.5 所示。

从以上介绍 date()函数的参数及图 9.5 所示的执行结果可见，使用 date()函数可以返回主机的当前日期。

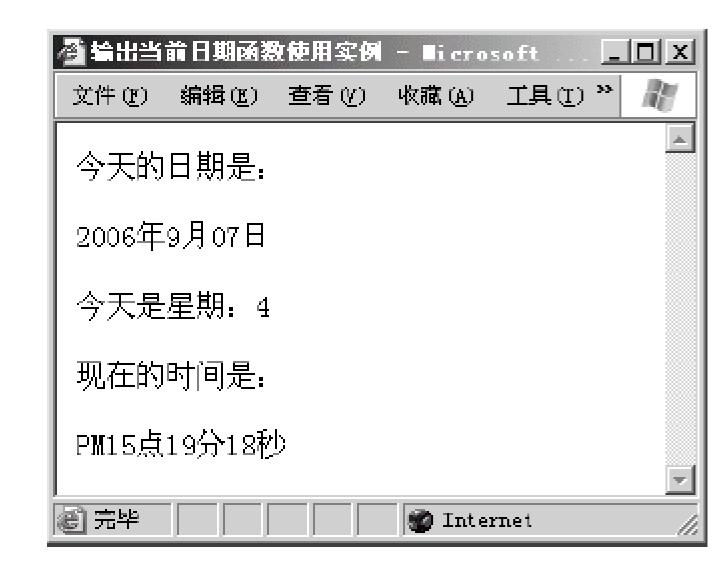


图 9.5　输出当前日期函数使用实例执行结果

9.3.3　获得时间及日期信息

通过 9.3.2 小节对 date()函数的介绍，知道在 PHP 中获得时间及日期信息是一件很容易的事情，做法就是执行 date()函数再加上不同的参数即可。下面介绍另一个常用的获得日期信息函数 getdate()。它的使用格式如下所示：

```
array getdate([int timestamp])
```

函数可包含参数，参数为一个表示日期的数字，如果没有参数则返回当前日期的信息。返回值为一个数组。其返回的数组中内容很多，返回数组的键名及相应值的情况请参见表 9.2。

表 9.2　getdate()函数的返回值数组的键名单元

键　名	说　明	返回值例子
"seconds"	秒的数字表示	0到59
"minutes"	分钟的数字表示	0到59
"hours"	小时的数字表示	0到23
"mday"	月份中第几天的数字表示	1到31
"wday"	星期中第几天的数字表示	0（表示星期天）到6（表示星期六）
"mon"	月份的数字表示	1到12
"year"	4位数字表示的完整年份	例如：1999或2003
"yday"	一年中第几天的数字表示	0~365
"weekday"	星期几的完整文本表示	Sunday~Saturday
"month"	月份的完整文本表示	January~December

下面继续通过实例向读者介绍 PHP 编程中如何获得时间及日期信息。

```
<html>
<head>
<title>获得时间及日期信息函数使用实例</title>
</head>
<body>
<?
$today=getdate();
echo "当前年份为：";
```

```
echo $today[year];
echo "<p>";
echo "当前月份为：";
echo $today[month];
echo "<p>";
echo "当前日期为：";
echo $today[mday];
echo "<p>";
echo "今天是全年中的第：";
echo $today[yday]."天";
echo "<p>";
echo "今天是星期：";
echo $today[wday];
echo "<p>";
?>
</body>
</html>
```

保存以上代码为“9-6.php”。在 PHP 执行环境中运行该文件，执行结果如图 9.6 所示。

除了获取当前日期之外，也可以在 getdate()函数中使用参数，参数可以为任意合法的、表示日期的数字，函数一样会返回这一日期的信息。

到这里 PHP 中的日期时间函数基本上就介绍完了。这些函数在平时编程过程中经常要用到。读者要熟练使用日期时间函数，为编写复杂的 Web 程序打下基础。

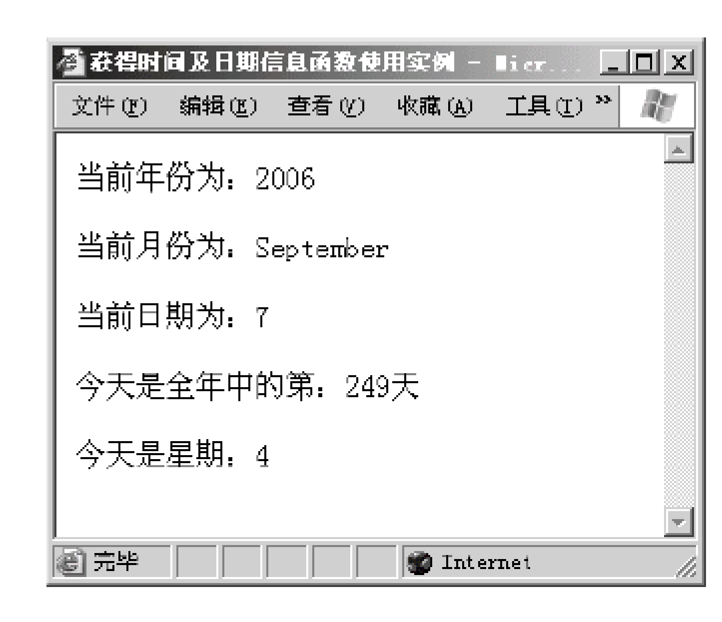


图 9.6　获得时间及日期信息函数使用实例执行结果

9.4　本 章 小 结

无论是通过系统内置变量获得主机信息，还是通过日期时间函数获得日期时间信息，都是编写大型 Web 应用程序中不可或缺的内容。在大型应用程序中都处于基础的位置。通过本章的学习，读者对如何获取主机相关信息，如何获取日期时间信息应该有了一个深刻的认识。

第 10 章

PHP 中的图像处理

- PHP 中的图像函数
- 使用 GD 库函数创建图像
- 使用 GD 库函数绘制图形
- GD 库函数综合应用实例

使用过 Internet 的用户都知道，一个网站不可能离开图片。各种各样的图片，会给死板的网页带来很多生机。PHP 是嵌入式语言，所以 PHP 页中除了 PHP 代码之外都是 HTML 标记，PHP 文件中也可以使用图片标记。但 PHP 又不仅仅只是能用图片而已，它还能对图片进行处理。这一章讲解 PHP 中的图像处理，内容包括 PHP 中的简单图像函数、GD 库的使用、图像使用实战、使用 GD 函数创建图像和使用 GD 库函数绘制几何图形。最后，本章还讲解了几个实际应用：添加图片文字水印、绘制直方统计图和制作图像的缩略图。

10.1　图 像 函 数

PHP 中有几个内置的简单的图像函数，还有功能丰富的 GD 库函数。简单的图像函数可以用来读取图像文件的长宽和图像的格式。而 GD 库函数功能就强大得多，可以创建图像、已有图像的缩略图、在图片文件上写字等，这一节就分别为读者做介绍。

10.1.1　访问图像的属性

在处理一个图像文件之前，了解图像的大小是很有必要的。PHP 中有这样一个函数用来返回图像的长、宽及格式。这个函数是 getimagesize()，它的使用格式如下所示：

```
array getimagesize(string filename [,array &imageinfo]);
```

函数的参数为指向图像文件路径及文件名的字符串，函数返回一个数组。其中数组有 4 个元素，第 1 个为图像文件的长，第 2 个为图像文件的宽，第 3 个为图像元素的类型。该函数支持多种类型的图像，其值类型如表 10.1 所示。

表 10.1　getimagesize()函数图像类型值及内容

函数返回值	指代的图像类型
1	GIF图像
2	JPG/JPGE图像
3	PNG图像
4	SWF（FLASH）文件
5	PSD（Photoshop）文件
6	BMP图像
7	TIFF
8	TIFF
9	JPC文件
10	JP2文件
11	JPX文件
12	JB2文件
13	SWC文件
14	IFF文件
15	WBMP文件
16	XBM文件

这些标记与 PHP 4.3.0 新加的 IMAGETYPE 常量对应。

第 4 个参数为文本字符串，内容为“height="yyy",width="xxx"”，可直接用于 IMG 标记。

下面通过一个实例来讲解如何使用 getimagesize()函数来访问图像的属性。具体内容参看以下代码：

```
<html>
<head>
<title>访问图像属性函数使用实例</title>
</head>
<body>
<?
$image="1.jpg";                              //定义指向图像文件的字符变量
echo "<img src=".$image.">";                 //插入图像
$temp=getimagesize($image);                  //使用函数并把返回值赋值给数组
echo "<p>";
echo "该图像的长为：";
echo $temp[0];                               //调用图像的长
echo "<p>";
echo "该图像的宽为：";
echo $temp[1];                               //调用图像的宽
echo "<p>";
echo "该图像的格式为：";
switch ($temp[2])                            //通过判断返回图像的格式
{
        case 1:                              //如果图像为 GIF
        echo "GIF 图像";
        break;
        case 2:                              //如果图像为 JPG
        echo "JPG/JPGE 图像";
        break;
        case 3:                              //如果图像为 PNG
        echo "PNG 图像";
        break;
        default:                             //除以上 3 种外的其他情况
        echo "未知图像格式";
        break;
}
?>
</body>
</html>
```

保存以上代码为“10-1.php”。然后，在 PHP 执行环境中运行该文件（注意在执行该文件以前，先把 1.jpg 放在 10-1.php 的同一个目录下），其执行结果如图 10.1 所示。

通过实例发现，getimagesize()函数能返回图像的长与宽及图像类型。这个函数对于网页中图像的插入是十分有用的。



图 10.1　访问图像属性函数使用实例执行结果

10.1.2　使用 GD 库函数

PHP 的 GD 库函数是专门用来处理图像、画图的库函数。这一节为读者介绍 PHP 中常见 GD 库函数的使用。在使用 GD 库之前要先让 PHP 解释器能加载 GD 库函数。方法就是修改 php.ini 文件。打开

php.ini 文件找到这一行：

```
;extension=php_gd.dll
```

把前面的“;”去掉，然后，保存 php.ini 文件即可。这样在启动 PHP 解释器时就会加载 GD 库函数。

在前面几章中已经向读者展示过 GD 库的强大功能。如第 6 章的例子 6-7.php，和第 7 章的例子 7-4.php。当时第 6 章的重点是介绍如何使用非系统库函数，而第 7 章主要是讲解三角函数。下面对这两个例子作详细的说明。

先来回顾第 6 章的例子 6-7.php：

```
<?
Header("Content-type: image/png");                        //输出一个 PNG 图片文件
$im=imagecreate(440,100);                                  //初始化图形区域
$black=imagecolorallocate($im, 0,0,0);                     //定义黑色
$white=imagecolorallocate($im, 255,255,255);               //定义白色
$yellow=imagecolorallocate($im,255,255,0);                 //定义黄色
$blue=imagecolorallocate($im,0,0,255);                     //定义蓝色
$red=imagecolorallocate($im,255,0,0);                      //定义红色
$zi=imagevolorallocate($im,255,0,255);                     //定义紫色
$color=$blue;                                              //定义$color 变量为蓝色
$font="wingdings.ttf";                                     //定义字体文件
imagefilledrectangle($im, 5, 5, 435, 95, $color);          //用蓝色画一个矩形
imagestring($im,5,7,10,"I:send",$white);                   //用白色写字符
for($i=0;$i<5;$i++)                                        //用循环画字符
    {
        imagettftext($im,40,0,90+$i*50,57,$yellow,$font,"Z");    //画出字符用黄色及字体
    }
imagestring($im,5,270,60,"to:YOU As a gift",$white);       //用白色写字符
imagestring($im,5,305,80,date(Y).".".date(m).".".date(d),$white);   //写出当前日期
Imagepng($im);                                             //创建图形
Imagedestroy($im);                                         //关闭图形
?>
```

该例子使用了 GD 库中的创建图像函数、画矩形函数、在图像上写字符函数等几个函数。执行结果详见第 6 章的图 6.7。

下面再来回顾一下第 7 章的例子 7-4.php：

```
<?
Header("Content-type: image/png");                         //输出文件头为 PNG 图片
$im = imagecreate(400,400);                                 //使用 GD 库函数创建区域
$black = imagecolorallocate($im, 0,0,0);                    //使用 GD 库函数定义黑色
$white = imagecolorallocate($im, 255,255,255);              //使用 GD 库函数定义白色
$yellow = imagecolorallocate($im,255,255,0);                //使用 GD 库函数定义黄色
$blue = imagecolorallocate($im,0,0,255);                    //使用 GD 库函数定义蓝色
$red = imagecolorallocate($im,255,0,0);                     //使用 GD 库函数定义红色
imagefilledrectangle($im, 5, 5, 395, 395, $yellow);         //使用 GD 库函数画矩形
for($i=1;$i<360;$i++)                                       //通过循环画点
{
    $temp=150*sin(2*(pi()/180)*$i);                         //通过三角函数计算值
```

```
        $x=$temp*cos((pi()/180)*$i)+200;                //通过三角函数计算点的横坐标
        $y=$temp*sin((pi()/180)*$i)+200;                //通过三角函数计算点的纵坐标
        imagesetpixel ($im,$x,$y,$red);                 //通过 GD 库函数画点
        $temp=150*cos(2*(pi()/180)*$i);                 //通过三角函数计算第 2 个值
        $x=$temp*cos((pi()/180)*$i)+200;                //通过三角函数计算点的横坐标
        $y=$temp*sin((pi()/180)*$i)+200;                //通过三角函数计算点的纵坐标
        imagesetpixel ($im,$x,$y,$blue);                //通过 GD 库函数画点
    }
    Imagepng($im);                                      //输出 PNG 图片
    Imagedestroy($im);                                  //清空图片
    ?>
```

该例除了使用创建图像函数之外，主要还使用了 GD 库函数中的画点函数。该函数的作用是在固定的位置画一个点。

下面附上 GD 库中的常用函数及意义，详见表 10.2。

表 10.2　常见 GD 库函数意义

gd_info	取得当前安装的GD库的信息
getimagesize	取得图像大小
image_type_to_extension	取得图像类型的文件后缀
image_type_to_mime_type	取得getimagesize，exif_read_data
exif_thumbnail，exif_imagetype	所返回的图像类型的MIME类型
image2wbmp	以WBMP格式将图像输出到浏览器或文件
imagealphablending	设定图像的混色模式
imageantialias	是否使用抗锯齿（antialias）功能
imagearc	画椭圆弧
imagechar	水平地画一个字符
imagecharup	垂直地画一个字符
imagecolorallocate	为一幅图像分配颜色
imagecolorallocatealpha	为一幅图像分配颜色+alpha
imagecolorat	取得某像素的颜色索引值
imagecolorclosest	取得与指定的颜色最接近的颜色的索引值
imagecolorclosestalpha	取得与指定的颜色加透明度最接近的颜色
imagecolorclosesthwb	取得与给定颜色最接近的色度的黑白色的索引
imagecolordeallocate	取消图像颜色的分配
imagecolorexact	取得指定颜色的索引值
imagecolorexactalpha	取得指定的颜色加透明度的索引值
imagecolormatch	使一个图像中调色板版本的颜色与真彩色版本更能匹配
imagecolorresolve	取得指定颜色的索引值或有可能得到的最接近的替代值
imagecolorresolvealpha	取得指定颜色+alpha的索引值或有可能得到的最接近的替代值
imagecolorset	给指定调色板索引设定颜色
imagecolorsforindex	取得某索引的颜色

续表

imagecolorstotal	取得一幅图像的调色板中颜色的数目
imagecolortransparent	将某个颜色定义为透明色
imageconvolution	用系数div和offset申请一个3x3的卷积矩阵
imagecopy	复制图像的一部分
imagecopymerge	复制并合并图像的一部分
imagecopymergegray	用灰度复制并合并图像的一部分
imagecopyresampled	重采样复制部分图像并调整大小
imagecopyresized	复制部分图像并调整大小
imagecreate	新建一个基于调色板的图像
imagecreatefromgd2	从GD2文件或URL新建一图像
imagecreatefromgd2part	从给定的GD2文件或URL中的部分新建一图像
imagecreatefromgd	从GD文件或URL新建一图像
imagecreatefromgif	从GIF文件或URL新建一图像
imagecreatefromjpeg	从JPEG文件或URL新建一图像
imagecreatefrompng	从PNG文件或URL新建一图像
imagecreatefromstring	从字符串中的图像流新建一图像
imagecreatefromwbmp	从WBMP文件或URL新建一图像
imagecreatefromxbm	从XBM文件或URL新建一图像
imagecreatefromxpm	从XPM文件或UR 新建一图像
imagecreatetruecolor	新建一个真彩色图像
imagedashedline	画一虚线
imagedestroy	销毁一图像
imageellipse	画一个椭圆
imagefill	区域填充
imagefilledarc	画一椭圆弧且填充
imagefilledellipse	画一椭圆并填充
imagefilledpolygon	画一多边形并填充
imagefilledrectangle	画一矩形并填充
imagefilltoborder	区域填充到指定颜色的边界为止
imagefilter	对图像使用过滤器
imagefontheight	取得字体高度
imagefontwidth	取得字体宽度
imageftbbox	给出一个使用FreeType2字体的文本框
imagefttext	使用FreeType2字体将文本写入图像
imagegammacorrect	对GD图像应用gamma修正
imagegd2	将GD2图像输出到浏览器或文件
imagegd	将GD图像输出到浏览器或文件
imagegif	以GIF格式将图像输出到浏览器或文件

续表

imageinterlace	激活或禁止隔行扫描
imageistruecolor	检查图像是否为真彩色图像
imagejpeg	以JPEG格式将图像输出到浏览器或文件
imagelayereffect	设定alpha混色标志以使用绑定的libgd分层效果
imageline	画一条线段
imageloadfont	载入一新字体
imagepalettecopy	将调色板从一幅图像复制到另一幅
imagepng	以PNG格式将图像输出到浏览器或文件
imagepolygon	画一个多边形
imagepsbbox	给出一个使用PostScript Type1字体的文本方框
imagepsencodefont	改变字体中的字符编码矢量
imagepsextendfont	扩充或精简字体
imagepsfreefont	释放一个PostScript Type1字体所占用的内存
imagepsloadfont	从文件中加载一个PostScript Type1字体
imagepsslantfont	倾斜某字体
imagepstext	用PostScript Type1字体把文本字符串画在图像上
imagerectangle	画一个矩形
imagerotate	用给定角度旋转图像
imagesavealpha	设置标记以在保存PNG图像时保存完整的alpha通道信息（与单一透明色相反）
imagesetbrush	设定画线用的画笔图像
imagesetpixel	画一个单一像素
imagesetstyle	设定画线的风格
imagesetthickness	设定画线的宽度
imagesettile	设定用于填充的贴图
imagestring	水平地画一行字符串
imagestringup	垂直地画一行字符串
imagesx	取得图像宽度
imagesy	取得图像高度
imagetruecolortopalette	将真彩色图像转换为调色板图像
imagettfbbox	取得使用TrueType字体的文本的范围
imagettftext	用TrueType字体向图像写入文本
imagetypes	返回当前PHP版本所支持的图像类型
imagewbmp	以WBMP格式将图像输出到浏览器或文件
imagexbm	将XBM图像输出到浏览器或文件
iptcembed	将二进制IPTC数据嵌入到一幅JPEG图像中
iptcparse	将二进制IPTC http://www.iptc.org/块解析为单个标记
jpeg2wbmp	将JPEG图像文件转换为WBMP图像文件
png2wbmp	将PNG图像文件转换为WBMP图像文件

从表 10.2 可以看出，PHP 的 GD 库函数功能相当丰富，几乎涵盖了图像处理中的所有内容。下面将通过几节内容为读者介绍常用的 GD 库函数的使用方法。

10.2　图像使用实战

从 10.1 节读者已经领会了 GD 库的强大功能。这一节将用 3 小节的内容介绍常用的 GD 库函数的使用方法。内容包括：如何创建图形，如何在图形上画图，绘制几何图形，在图形上写字等。从中读者能够学习到常用 GD 库函数的使用及图像处理的技巧。

10.2.1　使用 GD 库函数创建图像

在使用一个图像之前必须要先创建它。通过查看表 10.2 中的 GD 库函数列表可以发现，要创建一个图像就使用 imagecreate()函数。它的使用格式如下所示：

```
resource imagecreate(int x_size,int y_size)
```

参数 x_size、y_size 为整型变量，内容为所创建图像的宽与高。该函数返回一个图像标识符，代表了一幅宽与高为 x_size，y_size 的空白图像。

注意： 由于要向浏览器输出一幅图像，所以使用此函数之前要使用 header("Content-type: image/png"); 并且在使用 header()函数之前不能有任何内容的输出。

下面使用一个实例来说明如何使用 imagecreate()函数，具体内容参看以下代码：

```
<?
header("Content-type:image/png");                                   //向浏览器输出文件头
$im=imagecreate(100,50);                                            //使用函数创建图像
$background_color = imagecolorallocate($im,255,255,255);            //设置背景色
$text_color = imagecolorallocate($im,233,14,91);                    //定义文本内容颜色
imagestring($im,1,5,5,"A Simple Text String", $text_color);         //写文本
imagepng($im);                                                      //输出 PNG
imagedestroy($im);                                                  //销毁图像
?>
```

把以上代码保存为“10-2.php”，在 PHP 执行环境中运行该文件，其执行结果如图 10.2 所示。



图 10.2　imagecreate()函数使用实例执行结果

通过该实例及其执行结果可见创建一个图像的过程。先创建图像，再在图像上进行操作。本例实现的效果是先创建一个图像，再把文本输出到已经存在的图像上。

10.2.2　创建图形并在上面画图

10.2.1 小节介绍了如何创建图形，图形创建后，就是如何操作的问题了。这一小节讲解如何在图形上面画出几何图形。GD 库函数能直接在图像上绘制直线、矩形、圆形、椭圆形等几何图形。下面来分别说明这几个函数的使用方法：

- bool imageline(resource image,int x1,int y1,int x2,int y2,int color)函数。参数 image 为一个已经创建的图形对象，参数 x1、y1、x2、y2 均为整型数；color 为表示颜色的整型数（以上参数在其他函数中有相似的意义不再重复）。执行函数，将会用 color 颜色画一条从 x1，y1 到 x2，y2 的直线。
- bool imagerectangle(resource image,int x1,int y1,int x2,int y2,int col)函数。参数内容与 imageline()函数相似。执行函数，将会用 color 颜色画一个矩形，矩形的左上角为 x1，y1；右下角坐标为 x2，y2。
- bool imagefilledrectangle(resource image,int x1,int y1,int x2,int y2,int color).。执行函数，将会画一个矩形，与上一个函数不同的是一个用 color 填充的矩形。
- bool imageellipse(resource image,int cx,int cy,int w,int h,int color)函数。参数 cx、cy、w、h 均为整型数。执行函数，将会画一个颜色为 color 的椭圆，该椭圆的左上角为 cx，cy，w、h 为椭圆的宽度和高度，如果两者相等画出的就是一个正圆形。
- bool imagefilledellipse(resource image,int cx,int cy,int w,int h,int color)函数。基本内容同上也是绘制椭圆，不同的是此函数画出的椭圆也是填充的实心图形。

介绍了以上几个函数，下面就通过实例来向读者说明如何使用这些函数绘制几何图形：

```
<?
header("Content-type:image/png");                              //向浏览器输出文件头
$im=imagecreate(500,500);                                      //使用函数创建图像
$black=imagecolorallocate($im, 0,0,0);                         //定义黑色
$white=imagecolorallocate($im, 255,255,255);                   //定义白色
$yellow=imagecolorallocate($im,255,255,0);                     //定义黄色
$blue =imagecolorallocate($im,0,0,255);                        //定义蓝色
$red=imagecolorallocate($im,255,0,0);                          //定义红色
$zi=imagecolorallocate($im,255,0,255);                         //定义紫色
$background_color = imagecolorallocate($im,255,255,255);       //设置背景色
imageline($im,10,10,350,10,$white);                            //用白色画直线
imagerectangle($im,20,20,200,100,$blue);                       //用蓝色画一个矩形
imagefilledrectangle($im,100,200,200,300,$yellow);             //用黄色画一个填充矩形
imageellipse($im,50,50,150,150,$zi);                           //用紫色画一个椭圆
imagefilledellipse($im,50,50,150,350,$white);                  //用白色画一个正圆形
imagepng($im);                                                 //输出 PNG
imagedestroy($im);                                             //销毁图像
?>
```

把以上代码保存为“10-3.php”。然后在 PHP 执行环境中执行该文件，其执行结果如图 10.3 所示。

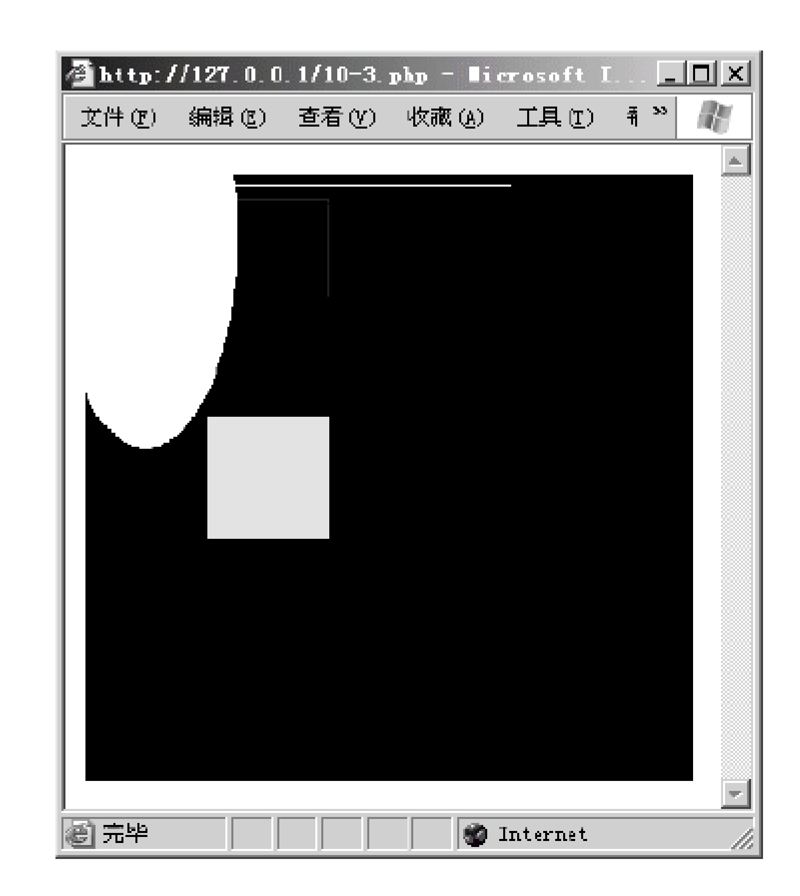


图 10.3　使用 GD 函数在图像上绘制几何图形

通过上面的实例及执行结果，可以发现 GD 库函数几乎可以绘制所有的几何图形。通过自己的创意，能绘出更多的项目。

10.2.3　绘制几何图形更多的探索

本节利用 GD 库中的画直线函数 imageline()来做更多的探索。利用画直线函数来达到画三角形、画五角星及更复杂图形的目的。

再复杂的图形归根结底都是由简单的直线或者曲线构成的，只要掌握了构图的规律，就可以用简单的直线来描绘出复杂的图形。下面的例子实现了用直线画正三角形。具体内容参见以下代码：

```
<?
header("Content-type:image/png");                              //向浏览器输出文件头
$im=imagecreate(400,300);                                      //使用函数创建图像
$white=imagecolorallocate($im, 255,255,255);                   //定义白色
$background_color = imagecolorallocate($im,255,255,255);       //设置背景色
imageline($im,100,50,300,50,$white);                           //画线
imageline($im,100,50,200,50+200*sin(pi()/3),$white);           //画线
imageline($im,200,50+200*sin(pi()/3),300,50,$white);           //画线
imagepng($im);                                                 //输出 PNG
imagedestroy($im);                                             //销毁图像
?>
```

保存以上代码为“10-4.php”。然后，在 PHP 执行环境中运行该 PHP 文件，其执行结果如图 10.4 所示。

既然能用画直线函数画出正三角形，当然也可以用它来绘制五角星了。下面的代码就实现了用画直线函数绘制五角星：

```
<?
header("Content-type:image/png");                              //向浏览器输出文件头
$im=imagecreate(400,400);                                      //使用函数创建图像
```

```
$black=imagecolorallocate($im, 0,0,0);                          //定义黑色
$white=imagecolorallocate($im, 255,255,255);                    //定义白色
$red=imagecolorallocate($im,255,0,0);                           //定义红色
$background_color = imagecolorallocate($im,255,255,255);        //设置背景色
imagefilledrectangle($im,5,5,395,395,$white);                   //用白色画一个矩形
$m=200;                                                         //边长
$n=sqrt($m*$m+$m*$m-2*$m*$m*cos(3*pi()/5));                     //大边长
$a=cos(2*pi()/5)*$n;                                            //设置变量以求顶点坐标
$b=sin(2*pi()/5)*$n;
$c=cos(pi()/5)*$m;
$d=sin(pi()/5)*$m;
imageline($im,200,50,200-$a,50+$b,$red);                        //画线
imageline($im,200,50,200+$a,50+$b,$red);                        //画线
imageline($im,200-$c,50+$d,200+$a,50+$b,$red);                  //画线
imageline($im,200+$c,50+$d,200-$a,50+$b,$red);                  //画线
imageline($im,200-$c,50+$d,200+$c,50+$d,$red);                  //画线
imagepng($im);                                                  //输出 PNG
imagedestroy($im);                                              //销毁图像
?>
```

保存以上代码为“10-5.php”。然后在 PHP 执行环境中运行该文件，执行结果如图 10.5 所示。

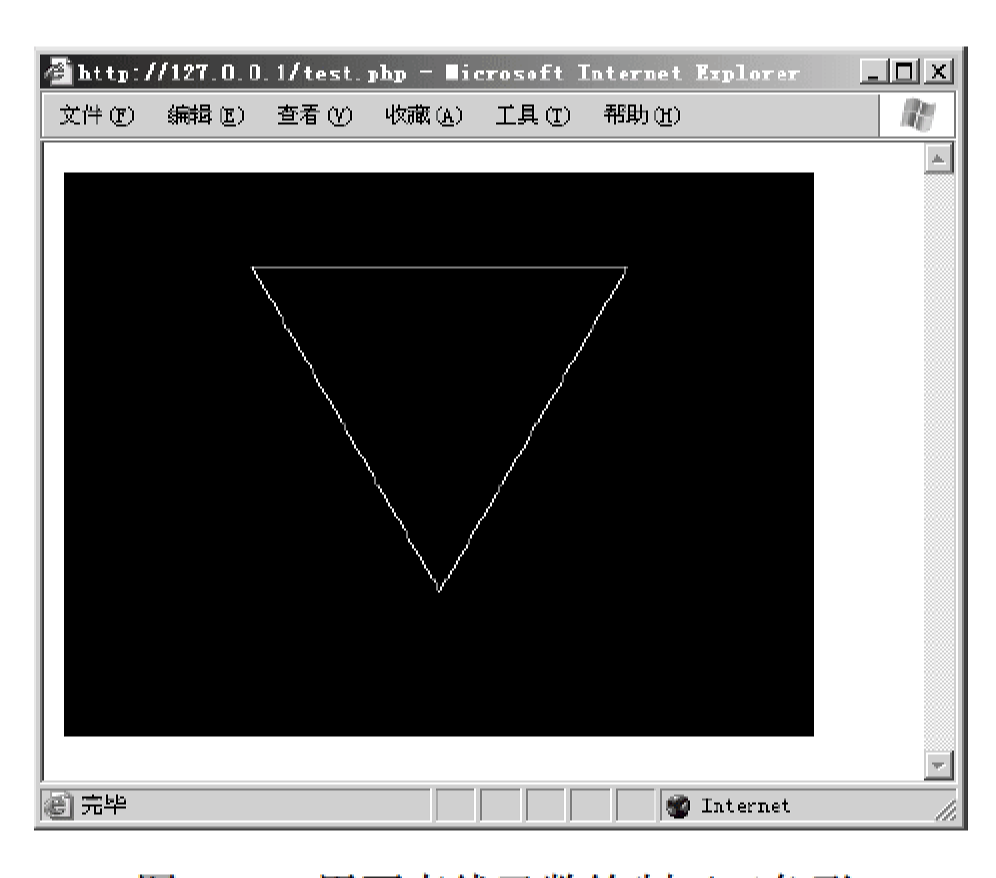


图 10.4　用画直线函数绘制正三角形

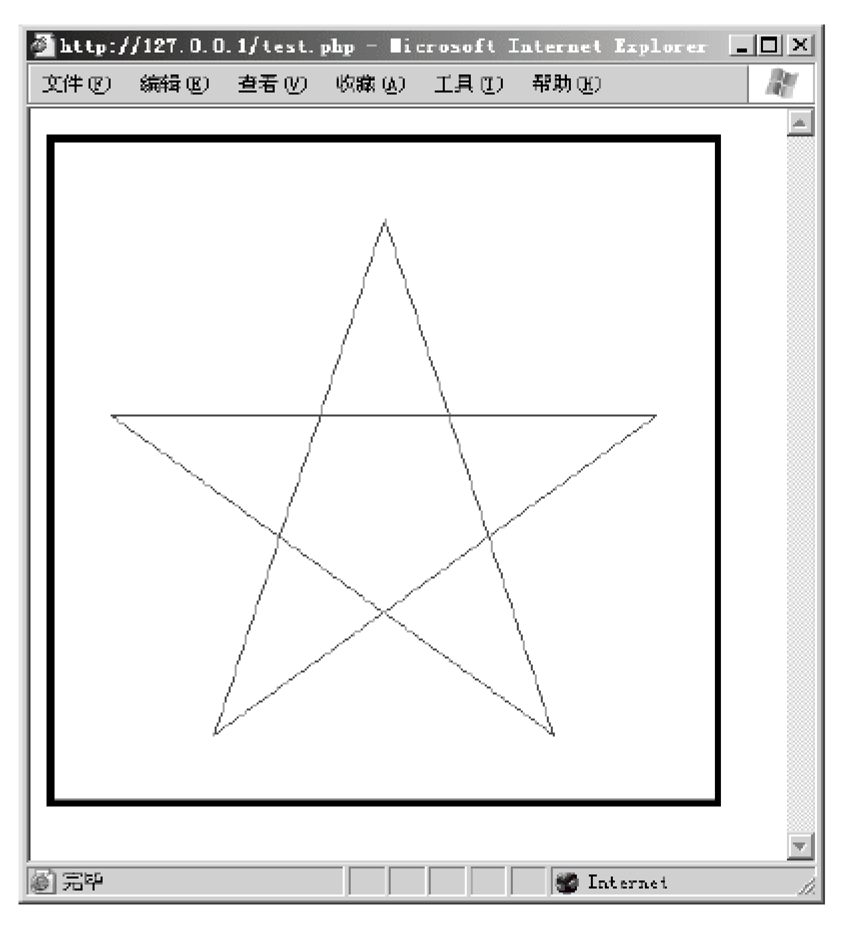


图 10.5　使用画直线函数绘制五角星

对于此五角星来说，画线的过程相当简单。主要是利用数学原理来求各个顶点的坐标相对麻烦。下面再通过一个例子，使用画线函数 imageline()来绘制正十二角星。具体内容参看以下代码：

```
<?
header("Content-type:image/png");                               //向浏览器输出文件头
$im=imagecreate(420,420);                                       //使用函数创建图像
$black=imagecolorallocate($im, 0,0,0);                          //定义黑色
$white=imagecolorallocate($im, 255,255,255);                    //定义白色
$blue=imagecolorallocate($im,0,0,255);                          //定义蓝色
$background_color = imagecolorallocate($im,255,255,255);        //设置背景色
imagefilledrectangle($im,5,5,415,415,$white);                   //用白色画一个矩形
```

```
imageline($im,210,10,110,210+100*sqrt(3),$blue);                //画线
imageline($im,210,10,310,210+100*sqrt(3),$blue);                //画线
imageline($im,210,410,110,210-100*sqrt(3),$blue);               //画线
imageline($im,210,410,110,210+100*sqrt(3),$blue);               //画线
imageline($im,10,210,210+100*sqrt(3),110,$blue);                //画线
imageline($im,10,210,210+100*sqrt(3),310,$blue);                //画线
imageline($im,410,210,210-100*sqrt(3),110,$blue);               //画线
imageline($im,410,210,210-100*sqrt(3),310,$blue);               //画线
imageline($im,110,210-100*sqrt(3),210+100*sqrt(3),310,$blue);   //画线
imageline($im,110,210+100*sqrt(3),210+100*sqrt(3),110,$blue);   //画线
imageline($im,210-100*sqrt(3),110,310,210+100*sqrt(3),$blue);   //画线
imageline($im,210-100*sqrt(3),310,310,210-100*sqrt(3),$blue);   //画线
imagepng($im);                                                  //输出 PNG
imagedestroy($im);                                              //销毁图像
?>
```

保存以上代码为“10-6.php”。在 PHP 执行环境下运行该文件，其执行结果如图 10.6 所示。

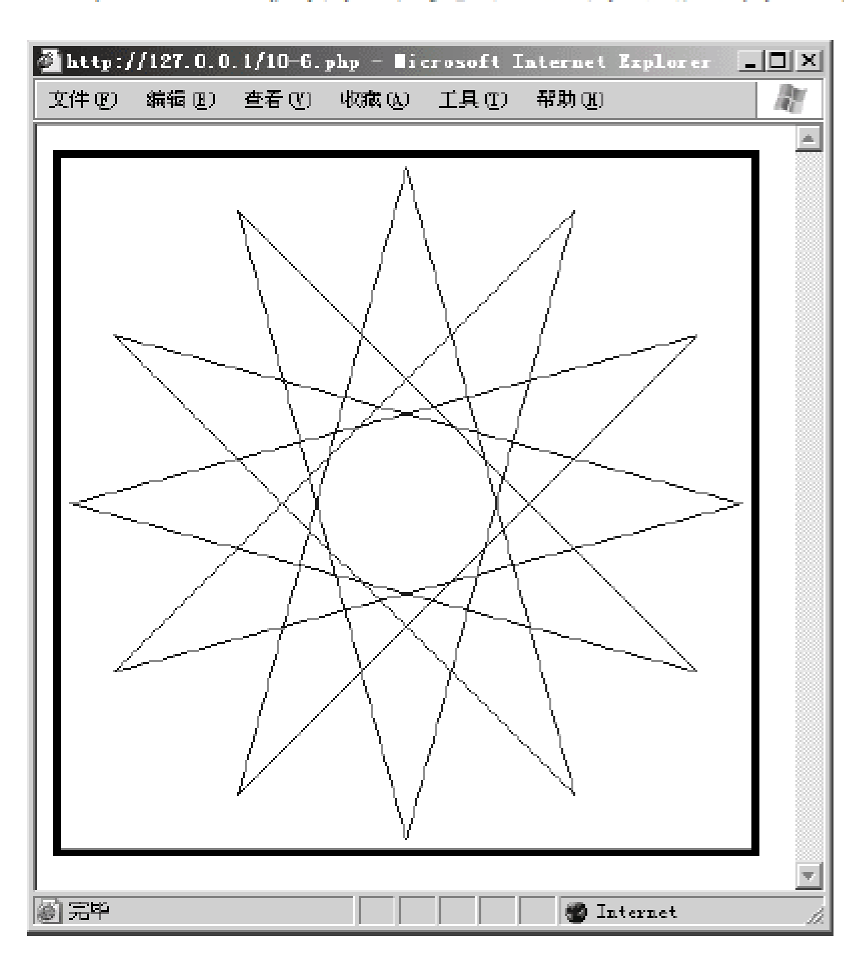


图 10.6　使用画直线函数绘制正十二角星

通过上面的 3 个例子，能够了解到利用 GD 库函数可以说是没有做不到的，只有想不到的。要善于利用库函数来实现特定的目的。

10.2.4　使用 GD 库函数在图片上写字

本节将使用 imagestring()函数在一张已经存在的图片上写字。比较常见的作用是当用户把图片上传到服务器后，通过 PHP 程序在图片上打上网站的标记，或者版权信息。该函数的使用格式如下所示：

```
bool imagestring(resource image,int font,int x,int y,string s,int col)
```

参数 image 为一个已经打开的图像对象，参数 x、y 为整型变量，内容为已经打开图像的坐标处。参数 font 为整型变量，内容为要显示的字体。如果要使用中文字体，则要把该字体文件放到程序同一个目录下。参数 string 为字符串类型，内容即为要写的文本内容。参数 col 为整型变量，内容为欲打印

文本的颜色。函数的返回值为布尔型变量，即如果成功打印文本则返回 True，否则就返回 False。

下面通过一个实例来说明如何使用 imagestring()函数在一张图片上写字。具体内容请参看以下代码：

```
<?
header("Content-type:image/png");                    //向浏览器输出文件头
$image="1.png";                                      //定义变量指向图像文件
$im=imagecreatefrompng($image);
$yellow=imagecolorallocate($im,255,255,0);           //定义黄色
$white=imagecolorallocate($im, 255,255,255);         //定义白色
imagestring($im,4,5,5,"I like this game!",$yellow);  //用黄色在图片上写字
imagestring($im,5,160,150,"TIAN ZHI HEN",$white);    //用白色在图片上写字
imagepng($im);                                       //输出 PNG
imagedestroy($im);                                   //销毁图像
?>
```

保存以上代码为“10-7.php”。然后在 PHP 执行环境中运行该文件（在执行该文件之前，要把“1.png”文件放到 10-7.php 的同一个目录下），其执行结果如图 10.7 所示。



图 10.7　GD 库函数在图片上写字

在使用 imagestring()函数时有一个问题需要注意，这个函数不支持中文，即 string 参数只能为英文或者数字。不过可以通过安装 Freetype 字体来解决这个问题。

10.2.5　使用 GD 库函数绘制直方统计图

从 10.2.3，10.2.4 小节读者学习了如何用 GD 库函数创建图像及如何在图片文件上输出字符。这一小节为读者介绍一个实用的例子，用 GD 库函数绘制直方统计图。具体内容参看以下代码：

```
<?
//首先定义一个数组，其内容可以表示为一个工厂全年生产效益
$num[0]=100;
$num[1]=120;
$num[2]=125;
$num[3]=130;
$num[4]=160;
$num[5]=200;
$num[6]=230;
$num[7]=250;
$num[8]=290;
$num[9]=310;
$num[10]=400;
$num[11]=200;
Header("Content-type: image/png");                   //输出头文件
$im=imagecreate(500,450);
$black=imagecolorallocate($im, 0,0,0);               //定义黑色
```

```
$white=imagecolorallocate($im, 255,255,255);                //定义白色
$yellow=imagecolorallocate($im,255,255,0);                  //定义黄色
$blue =imagecolorallocate($im,0,0,255);                     //定义蓝色
$red=imagecolorallocate($im,255,0,0);                       //定义红色
imageline($im,5,5,5,435,$white);                            //画出纵坐标
imageline($im,5,435,400,435,$white);                        //画出横坐标
for($i=0;$i<count($num);$i++)                               //循环画出直方图
{
    imagefilledrectangle($im,($i+1)*30,440-$num[$i]-5,($i+1)*30+20,435,$color);
}
for($i=0;$i<count($num);$i++)                               //循环画出数值
{
    imagestring($im,4,($i+1)*30,440-$num[$i]-5,"$num[$i]",$blue);
}
for($i=1;$i<13;$i++)                                        //循环画出横坐标单位
{
    imagestring($im,4,$i*30,430,"$i",$red);
}
for($i=0;$i<5;$i++)                                         //循环画出纵坐标单位
{
    $s=$i*100;
    imagestring($im,4,5,435-$s,"$s",$white);
}
Imagepng($im);                                              //创建图像
Imagedestroy($im);                                          //关闭图像
?>
```

保存以上代码为“10-8.php”。然后在 PHP 执行环境中运行该文件，其执行结果如图 10.8 所示。

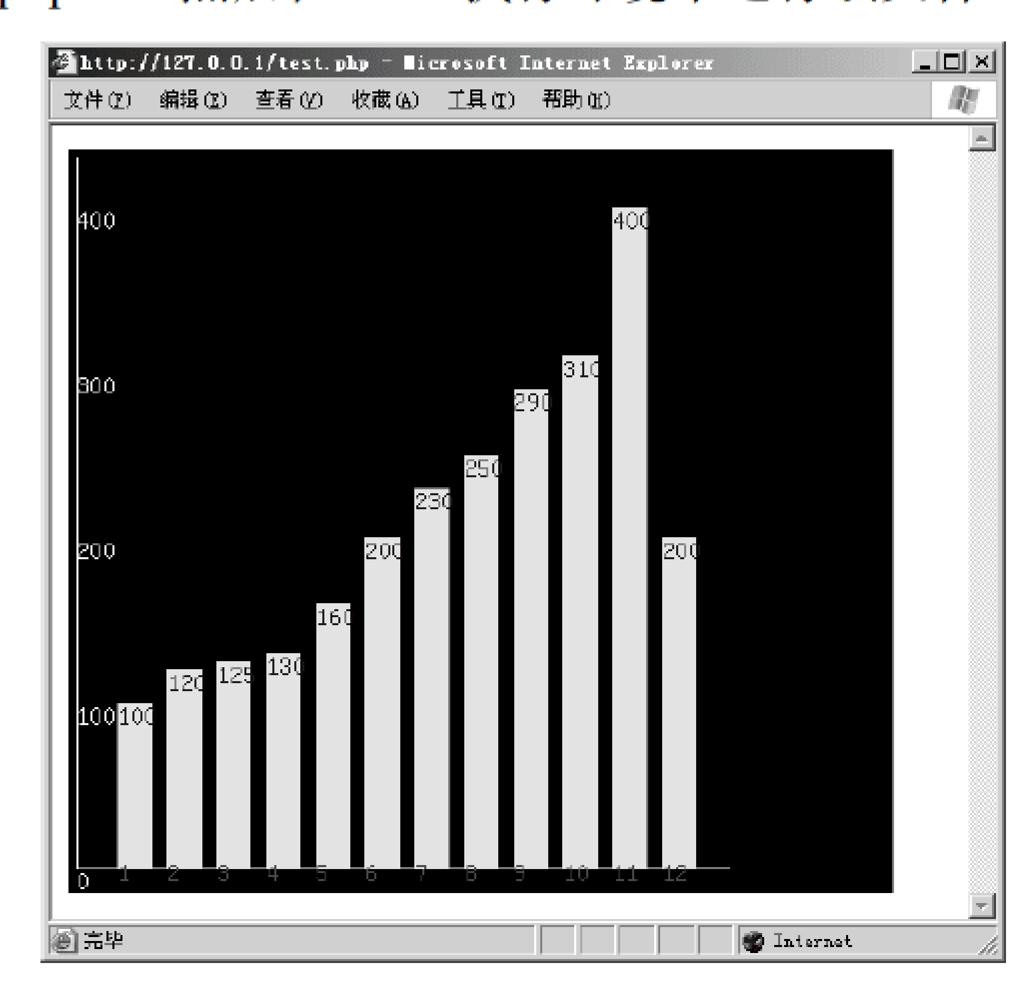


图 10.8　使用 GD 库函数绘制直方统计图使用实例执行结果

通过以上实例及执行结果可以看出 GD 库函数配合数学函数可以画出功能强大的数学图形，可以把数字内容变为直观的图像。这样可以增加程序的亲和力，使原本枯燥的数字更容易使用户接受，从而增加程序的易用性。

10.2.6　用 GD 库函数创建图像的缩略图

在编写 Web 应用程序时，要通盘考虑全面的内容，如网页加载时打开的图像太大，就会延长网页打开的速度，从而会使有的用户失去耐心。如果在用户打开图片前，生成欲打开图像的缩略图让用户了解大致情况，就会好得多。GD 库函数就有创建图像缩略图的功能。这一节就来介绍如何使用 GD 库函数创建一幅图像的缩略图。

在介绍程序之前，先来认识 GD 库函数 imagecopyresized()。它的使用格式如下所示：

```
int imagecopyresized(int dst_im,int src_im,int dstX,int dstY,int srcX,int srcY,int dstW,int dstH,int srcW,int srcH);
```

该函数可以复制新图，并重新调整图片的大小尺寸。参数都是目的在前，来源在后。参数 dst_im 及 src_im 为图片的句柄。参数 dstX、dstY、srcX、srcY 分别为目的及来源的坐标。参数 dstW、dstH、srcW、srcH 分别为目的及来源的宽及高，若欲调整新图的尺寸就在这里设定。

可见，通过此函数，就可以达到调整图片大小的目的，生成缩略图自然也是可以的。下面就通过一个实例来向读者作一介绍。具体内容参见以下代码：

```
<?
function resizeimage($srcfile,$rate=.5)                          //建立一个函数以指定比率缩放图像
{
    $size=getimagesize($srcfile);                                //判断图片类型及大小
    switch($size[2])                                             //判断图片类型根据类型创建图片
    {
        case 1:
        $img=imagecreatefromgif($srcfile);
        break;
        case 2:
        $img=imagecreatefromjpeg($srcfile);
        break;
        case 3:
        $img=imagecreatefrompng($srcfile);
        break;
    }
    $srcw=imagesx($img);                                         //源图片的宽度
    $srch=imagesy($img);                                         //源图片的高度
    $dstw=floor($srcw*$rate);                                    //目的图片的宽度
    $dsth=floor($srch*$rate);                                    //目的图片的高度
    $im=imagecreate($dstw,$dsth);                                //新建一个真彩色图像
    $black=imagecolorallocate($im,255,255,255);                  //定义黑色
    imagefilledrectangle($im,0,0,$dstw,$dsth,$black);            //画一个图色的矩形
    imagecopyresized($im,$img,0,0,0,0,$dstw,$dsth,$srcw,$srch);  //重新定义新图像
    imagejpeg($im);                                              //以 JPG 格式输出新图像
    imagedestroy($im);                                           //释放源图像
    imagedestroy($img);                                          //释放目的图像
}
$im1="1.jpg";                                                    //定义变量
resizeimage($im1);                                               //调用函数参数$rate 使用默认参数
?>
```

保存以上代码为“10-9.php”。然后在 PHP 执行环境中执行以上代码（注意要把“1.jpg”放到 10-6.php 的同一个目录下），执行结果如图 10.9 所示。



图 10.9　用 GD 库函数创建图像的缩略图

通过以上输出的图像，与实际使用的“1.jpg”的比较，发现程序正确输出了源图像的缩略版，达到了创建缩略图的使用目的。

从上面的函数也可以看出，通过此函数不仅可以创建缩略图，而且可以实现图像的任意大小的缩放。可见 GD 库函数功能的强大。

10.3　本章小结

这一章为读者介绍了 PHP 中与图像相关的函数及 GD 库函数，重点介绍了 GD 库函数。通过几个实例学习了几个常用的 GD 库函数。本章知识点有 PHP 中的获取图像信息函数 getimagesize()、GD 库函数的使用、用 GD 函数创建图像、使用 GD 库函数绘制几何图形、使用 GD 库函数在图片上写字和两个综合实例，即用 GD 库函数创建直方统计图和用 GD 库函数创建一幅图像的缩略图。通过本章的学习，用户对 GD 库函数有了一个初步的了解。更多更深层次的学习，需要在运用所学知识解决实际问题过程中领会。

第 11 章 PHP 中的 Session 与 Cookie

- 如何使用 Session
- 使用 Session 时的注意事项
- 如何使用 Cookie
- Cookie 综合使用实例

编写 Web 互动程序，保存用户的登录信息是十分必要的，这样可以提高网站对用户的吸引力。目前常见的存储机制有两种：一种是把用户信息存储于服务器端的主机上的 Session；另一种是把用户信息存储于客户端即客户机上的 Cookie（小甜饼）。这两种机制各有各的特点，这一章就来为读者介绍 PHP 中的 Session 和 Cookie。本章内容包括：Session 的使用注意事项、有关的 Session 函数及它们的使用方法、Cookie 的使用、PHP 中的 Cookie 相关函数及使用、为 Cookie 设置生命期和 Cookie 综合运用实例。

11.1　Session 的使用

Session 会话适合于存储用户的信息量比较少的情况。如果用户需要存储的信息量相对较少，并且对存储内容不要求长时间存储时，使用 Session 把信息存储于服务器端会比较适合。Session 会话允许用户注册任意数目的变量并保留给各个请求使用。当用户访问网站时，PHP 会自动（如果 php.ini 中的 session.auto_start 被设为 1）或在用户请求时（由 session_start()明确调用或 session_register()暗中调用）检查请求中是否发送了特定的会话 ID。如果是，则之前保存的环境就被重建。这一节就为读者介绍 PHP 中 Session 的使用。

11.1.1　如何使用 Session

PHP 中有多个 Session 函数来实现 Session 的定义、存储、更改、删除等操作。有了这些函数，使用 Session 就变得相当简单了。这一小节就为读者来介绍 PHP 中的 Session 函数及它们的使用。PHP 中常用的 Session 函数，详见表 11.1。

表 11.1　常用的 Session 相关函数

函　数　名	作　　用	参数及返回值
session_start(void)	此函数初始化一个新的Session，若该客户已在Session之中，则连上原有的Session	该函数没有参数，且返回值均为True。需要注意的一点是，在session_start()函数之前不能有任何的内容输出，否则就会发生错误
session_destroy(void)	此函数结束目前的Session	与session_start()函数一样，session_destroy()函数也没有参数，且返回值也均为True
session_name(string [name])	本函数可取得或者重新设定目前Session的名称	若无可选参数name则表示单单获取目前Session名称，加上参数则表示将Session名称设为参数name
session_module_name(string [module])	本函数可取得或者重新设定目前Session的模组	若无可选参数module则表示只获取目前Session的模组，加上参数则表示将Session模组设为参数module
session_save_path(string [path])	本函数可取得或者重新设定目前存放Session的路径	若无可选参数path则表示只有取得目前Session的路径目录名，加上参数path则表示将Session存在新的路径上
session_id(string [id])函数	本函数可取得或者重新设定目前存放Session的代号	若无参数id则表示只有取得目前Session的代号，加上参数则表示将Session代号设成新指定的id。输入及传回均为字串
session_register(string name)	本函数在全局变量中增加一个变量到目前的Session之中	参数name为欲检查的变量名。如果注册则返回True，反之则返回False

续表

函　数　名	作　　用	参数及返回值
session_unregister(string name)	本函数与session_register()函数相反为，功能是在目前的Session之中删除全局变量上的变量	参数name即为欲删除的变量名。成功则返回True值，反之则返回False
session_is_registered(string name)	本函数可以检查某个变量名是否已经被注册为Session	参数name为欲检查的变量名。如果注册则返回True，反之则返回False
session_decode(string data)	本函数可将Session内的资料解码	参数data即为欲解码的资料。成功则返回True值，否则返回False
session_encode(void)	本函数可将Session资料编码，编码以ZEND引擎做杂凑编码	本函数没有参数。成功则返回True值，否则返回False

下面通过一个实例来说明如何在 PHP 中使用 Session，具体内容参看以下代码。以下代码实现了判断 Session 变量是否被注册，注册 Session 变量和显示 Session 变量的功能：

```
<?
session_start();                              //开始使用 session
?>
<html>
<head>
<title>session 使用实例</title>
</head>
<body>
<?
$username1="guest";                           //定义变量
if(!session_is_registered("username"))        //判断 session 是否被注册
{
    session_register("username");             //注册 session
    $_session["username"]=$username1;         //给 session 变量赋值
}
echo $_session["username"];                   //显示 session
session_destroy();                            //结束 session
?>
</body>
</html>
```

保存以上代码为“11-1.php”。在 PHP 执行环境中运行该文件（此时的 register_globals=on），其执行结果如图 11.1 所示。

通过上面的例子及执行结果可以发现，程序正确的执行了相关操作。它的运行机制为程序先判断名字为“username”的 Session 有没有被注册，如果没有则注册 Session 变量，并赋值。然后把名字为“username”的 Session 变量的内容显示出来。

图 11.1　session 使用实例执行结果

11.1.2　Session 使用实例

上面的例子只是 Session 最简单的应用，实际上 Session 能做的事情远不止于此，它能存储多种信息，结合程序能做更多的事情，如以下代码就使用 Session 实现了一个简单的站点计数器：

```
<?
session_start();                               //开始使用 Session
if(!session_is_registered('count'))            //如果没有注册 Session
{
    session_register('count');                 //注册 Session
    $count=1;                                  //count 变量为 1
}
else                                           //如果已经注册 Session
{
    $count++;                                  //变量在原有基础上增加 1
}
?>
<html>
<head>
<title>使用 session 的网页计数器</title>
</head>
<body>
<p>
你好，你已经浏览本网页 <?php echo $count; ?>次
</p>
</body>
</html>
```

保存该代码为“11-2.php”文件。然后在 PHP 执行环境中运行该文件，其执行结果将会如图 11.2 所示。

从图 11.2 中可以了解到，该页面注册了$count 变量为 Session。并且随着刷新，$count 的数量也随之增加，这样就实现了统计浏览次数的目的。

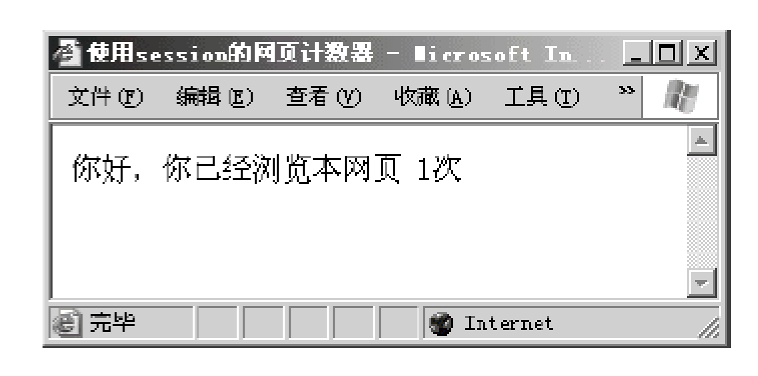


图 11.2　使用 session 的网页计数器执行结果

11.1.3　使用 Session 的注意事项

在使用 Session 时有下列事项需要引起注意：

- 如果在 PHP.ini 文件中启用了 session.auto_start，就不能将对象放入会话中，因为类定义必须在启动会话之前加载以在会话中重建对象。
- 请求结束后所有注册的变量都会被序列化。已注册但未定义的变量被标记为未定义。在此后的访问中这些变量也不会被会话模块定义，除非用户以后定义它们。
- 另外有些类型的数据不能被序列化，因此也就不能保存在会话中。包括 resource 对象类型变

量或者有循环引用的对象（即某对象将一个指向自己的引用传递给另一个对象）。

- 默认情况下，所有与特定会话相关的数据都被存储在由 php.ini 文件的 session.save_path 选项所指定的目录下的一个文件中。对每个会话建立一个文件（不论是否有数据与该会话相关）。这是由于每打开一个会话即建立一个文件，不论是否有数据写入到该文件中。注意由于和文件系统协同工作的限制，此行为有个副作用，有可能造成用户定制的会话处理器（例如用数据库）丢失未存储数据的会话。
- 如果 php.ini 的配置没有开启 register_globals，则只可以用_SESSION 数组来引用 Session 变量。如，注册的$var 变量只能这样引用“$_SESSION["var"]”。而如果开启了 register_globals 选项，则可以直接使用$var。
- 由于 Cookie 存在于客户端，所以它可以被设置为长时间甚至是无限期存在。而 Session 因为是存储在服务器端，所以它的生命期很短。注册过的变量过段时间就会自动失效。

11.2　Cookie 的使用

11.1 节给读者介绍了 PHP 中关于 Session 的使用，这一节来为读者介绍 PHP 中 Cookie 的使用。通过本节的学习，将会使读者对 PHP 中的 Cookie 的使用有一个深刻的理解。并且能够使用 Cookie 去实际处理一些问题。

11.2.1　为什么使用 Cookie

同样作为互动 Web 页的存储机制 Session 与 Cookie 有其相似之处。那为什么在有了 Session 后还要使用 Cookie 呢？因为同 Session 相比 Cookie 有其独特之处。下面就来说明一下二者的异同，比较过二者的异同后，读者就会明白为什么要用 Cookie 了。

- Cookie 保存在客户端，而 Session 的内容保存在服务器端，只是把一个 session id 保存在客户端。也就是说，Cookie 比 Session 更安全，同时也可以说明 Session 是基于 Cookie 的（用 Cookie 来保存 session id）。
- 由于 Session 是保存在服务器端的，因此会占用服务器的空间，所以在一定时间内如果没有活动时 Session 就会过期，而且在浏览器关闭后也会作废。而 Cookie 就没有这方面的限制，它可以设定一个比较长的过期时间。
- 每个浏览器只能保存同一个域名下的至多 20 个 Cookie，而且每个 Cookie 的大小不能超过 4KB，而 Session 就没有这些限制（因为保存在服务器端）。
- 有些人会禁止浏览器接受 Cookie，而此时 Session 还是可以用的（注意 session 也可以不通过 Cookie 来实现）。

通过以上比较可以发现：Cookie 与 Session 各有各的用处，各有各的特点，两者不能够相互替代。11.2.2 小节将来了解一下，在 PHP 中如何使用 Cookie。

11.2.2　怎样使用 Cookie

在 PHP 中使用 Cookie 很简单，简单到只用一句函数就可以注册 Cookie 变量，这个函数是 setcookie()。其使用格式如下所示：

```
bool setcookie(string name[string value[int expire[string path[string domain[bool secure[bool httponly]]]]]] )
```

本函数会跟着文件头信息 header 送出一段信息字串到浏览器。由于是同 header 一起发送，所以该函数同其他 header 函数一样，在执行前不能有任何的 HTML 输出。实际上 Cookie 也算标头的一部份。

setcookie()函数的参数除了第 1 个 name 之外，都是可以省略的。参数 name 表示 Cookie 的名称；value 表示这个 Cookie 的值，value 参数为空字串则表示取消浏览器中名称为 name 的 Cookie 的值；expire 表示该 Cookie 的有效时间；path 为该 Cookie 的相关路径；domain 表示 Cookie 的网站；secure 则需在 https 的安全传输时才有效。

下面用一个实例来说明如何在 PHP 中使用 setcookie()函数来定义并使用 Cookie。具体内容参见以下代码。下面为文件“11-3.php”的内容：

```
<?
$username=$_GET["username"];                                //通过 URL 获得参数
if(!$_GET["username"])                                      //如果没有参数执行内容
{
setcookie("username","");                                   //取消 username 的资料
echo "没有指定用户名！";
echo "<p>";
echo "或者用户名不存在！";
}
else                                                        //如果存在参数
{
    setcookie("username","$username");                      //注册用户名
    echo "注册成功，点<a href=11-4.php>这里</a>查看";       //显示链接以查看 Cookie 信息
}
?>
<html>
<head>
<title>注册用户信息</title>
</head>
<body>
</body>
</html>
```

保存下面的代码为“11-4.php”文件：

```
<html>
<head>
<title>显示用户信息</title>
</head>
<body>
```

```
<?
echo "注册用户名为：";
echo $_COOKIE[username];                    //显示 Cookie 信息
?>
</body>
<html>
```

分别保存以上两个文件为“11-3.php”与“11-4.php”。然后在 PHP 执行环境中运行 11-3.php，不带任何参数，其执行结果如图 11.3 所示。

如果加上“username”参数，如“11-3.php?username=guest”再次运行，效果就会不一样。其执行结果如图 11.4 所示。

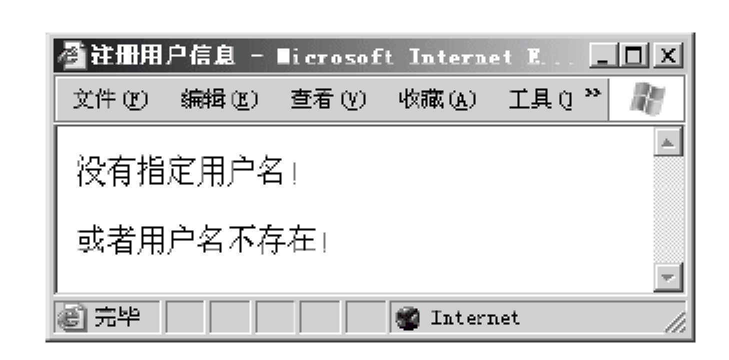


图 11.3　不带参数执行 11-3.php 的执行结果

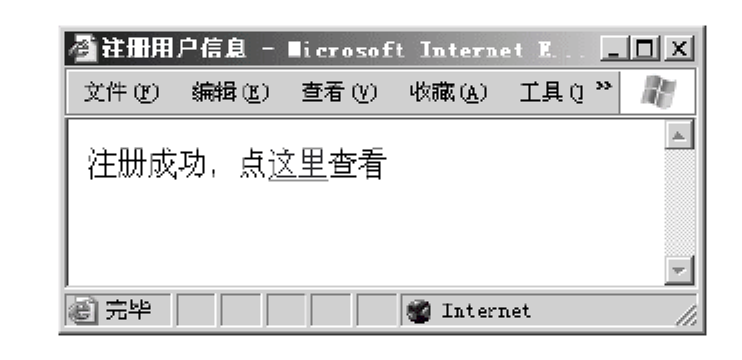


图 11.4　加上参数后执行 11-3.php 的执行结果

这里已经把 username 注册为 Cookie 其值为“guest”，然后单击“这里”链接，将会出现 11-4.php 的执行情况，如图 11.5 所示。

图 11.5 显示出了所注册的用户名。说明 Cookie 已经设置成功。由于没有给 Cookie 设置生命期，所以该 Cookie 的值会随着浏览器的关闭而中止。如果要保持 Cookie 的值长期有效，为其设置生命期是有必要的。11.2.3 小节将为读者讲解。



图 11.5　11-4.php 的执行结果

11.2.3　设置 Cookie 生命期

在实际使用 PHP 编写 Web 应用程序时，为了方便用户，需要对 Cookie 设置生命期。有的论坛在登录时就会有个选择，保存用户登录记录，并且还可以设置时间，如，保存一个月、三个月、甚至是一年。

其实保存用户登录记录的实质就是保存了用户的登录信息 Cookie 到用户客户机上。并且为该 Cookie 设置了较长的生命期，这样就避免了用户再次登录时，需要重新输入的问题。

如何为 Cookie 设置生命期？从 11.2.2 小节中讲到 setcookie()函数时就提到了这个问题，就是函数的第 3 个参数 expire，它的作用就是为 Cookie 设置生命期。如果这一项为空，就说明为最短生命期，即随着浏览器的关闭而中止。那么只要带有 expire 参数，就说明要为 Cookie 设置生命期，其单位为日期和时间。一般采用这种样式：

```
time()+N
```

其中的 time()表示当前时间，后面的 N 表示秒数。time()+N 的意思就是在当前时间后多少秒过期。如果要设置过期时间为一天就这样表示：

```
setcookie(name,value,time()+60*60*24);
```

其中的 60*60*24 即表示一天。因为一天有 24 小时，一小时有 60 分钟，一分钟有 60 秒。以上代码即表示设置 name 的值为 value 的 Cookie 变量有效期为从当前时间起 24 小时，即一天一夜。

知道了如何设置一天，那么，设置一月，甚至一年都可以触类旁通了。如设置有效期为一月就这样表示：

```
setcookie(name,value,time()+60*60*24*30);
```

同理，设置一年有效期就要像这样：

```
setcookie(name,value,time()+60*60*24*30*365);
```

至此，相信读者对如何为 Cookie 设置生命期，都有了一个深刻的理解。11.2.4 小节将为读者介绍一个综合应用的实例，来实际操作一下 Cookie。

11.2.4　Cookie 综合应用实例——网页风格转换

通过前几小节的介绍，读者对 PHP 中如何使用 Cookie 有了一个比较深刻的了解，这一节，将使用一个综合的例子来向读者具体说明在 PHP 中如何使用 Cookie。

Cookie 常用于用户登录记录、相关设置记录等。这一节的例子就来为读者说明这个问题。具体内容参看以下代码，以下代码为 11-5.php 前台界面代码：

```
<html>
<head>
<title>用户登录前台</title>
</head>
<body topmargin="100">
<center>
<table border=1>
<form method=post action=11-6.php>
<tr>
<td>请输入用户名：</td>
<td><input type=text name=username size=20></td>
</tr>
<tr>
<td>请输入密码：</td>
<td><input type=password name=password size=20></td>
</tr>
<tr>
<td>请选择保存期限：</td>
<td>
<select name=time size=1>
<option value=1>不保存
<option value=2>保存一天
<option value=3>保存一月
<option value=4>保存一年
</select>
</td>
```

```
</tr>
<tr>
<td colspan=2><input type=submit value="确定"></td>
</tr>
</form>
</table>
</center>
</body>
<html>
```

下面的代码为 11-6.php 后台执行代码。

```
<?
$username=$_POST["username"];                          //通过 POST 获得参数
$time=$_POST["time"];                                  //通过 POST 获得 time 变量
if(!$_POST["username"])                                //如果没有参数执行内容
{
    echo "没有输入用户名";
    echo "<p>";
    echo "点<a href=11-5.php>这里</a>返回";
}
else                                                   //如果存在参数
{
    switch $time                                       //判断有效期
    {
        case 1:
        $time=time();
        break;
        case 2:
        $time=time()+60*60*24;
        break;
        case 3:
        $time=time()+60*60*24*30;
        break;
        case 4:
        $time=time()+60*60*24*30*365;
        break;
    }
    setcookie("username","$username",$time);           //注册用户名
}
?>
<html>
<head>
<title>注册用户信息</title>
</head>
<body>
<?
echo "注册用户名为：";
echo $_COOKIE["username"];
echo "Cookie 有效期为：";
```

```
switch $time
{
        case 1:
        echo "不保存";
        break;
        case 2:
        echo "一天";
        break;
        case 3:
        echo "一月";
        break;
        case 4:
        echo "一年";
        break;
}
?>
</body>
</html>
```

分别保存以上代码为“11-5.php”与“11-6.php”。然后先在 PHP 执行环境中运行 11-5.php，执行结果如图 11.6 所示。

然后，按照上面内容输入用户名、密码（因为只是测试 Cookie，所以其实这两项都是虚拟的，并不存在用户数据库，只要输入内容就行）。并选择相应的 Cookie 保存期限，然后单击“确定”按钮，程序开始启用 11-6.php 的后台操作。其结果如图 11.7 所示。



图 11.6　11-5.php 执行结果



图 11.7　后台处理后的执行结果

11-5.php 及 11-6.php 就是 Cookie 用于简单的用户注册登录，并保存用户记录的情况。下面再介绍一个实例，用 Cookie 保存用户选择的网站风格。某些网站或者论坛具有多种风格供用户选择，这样可以满足用户不同的需求。

下面就介绍如何用 Cookie 来改变网站的风格。具体内容参看以下代码。

先来编写只有一句 PHP 代码的 11-7.php 文件，该文件中大部分内容均为标准的 HTML 内容：

```
<html>
<head>
```

```
<title>用 cookie 实现改变网站风格</title>
<?
echo "<LINK href='style".$_COOKIE["style"].".css' rel=stylesheet>";        //根据 Cookie 使用样式
?>
</head>
<body>
<table>
<p>用 Cookie 实现改变网站风格</p>
<tr>
<td><a href=11-8.php?style=1>风格 1</a></td>
<td><a href=11-8.php?style=2>风格 2</a></td>
<td><a href=11-8.php?style=3>风格 3</a></td>
<td><a href=11-8.php>无样式</a></td>
</tr>
<tr>
<td colspan=4><h1>慧能</h1></td>
</tr>
<tr>
<td colspan=4><h2>菩提本无树</h2></td>
</tr>
<tr>
<td colspan=4><h2>明镜亦非台</h2></td>
</tr>
<tr>
<td colspan=4><h2>本来无一物</h2></td>
</tr>
<tr>
<td colspan=4><h2>何处惹尘埃</h2></td>
</tr>
</table>
</body>
</html>
```

下面来编写处理提交样式的 PHP 文件 11-8.php。其功能是根据提交内容注册相应的 Cookie。并在处理后返回 11-7.php：

```
<?
$style=$_GET["style"];                              //判断参数来源
setcookie("style","$style",time()+60*60*24);        //根据参数注册 Cookie
?>
<html>
<head>
<title>正在处理</title>
</head>
<meta http-equiv="refresh" content="0; url=11-7.php">
<body>正在处理
</body>
```

另外在使用这个实例之前还要编写相应的 CSS 文件，并命名为“style1.css”、“style2.css”、“style3.css”。这 3 个 CSS 文件的内容如下所示（作相应保存即可）。

style1.css：

```
BODY, TD {
		font : 400 normal 100% Arial, Helvetica, sans-serif;
}
A {
	color: #FFD700;
	text-decoration: none;
	background: Black;
	width: 80px;
	padding: 3px;
}
A:hover
{
	background: Maroon;
}

H1 {
	background: #228B22 none;
	color: #FF6347;
	font-size: 150%;
	text-align: right;
}
H2 {
	background : #FFDEAD none;
	color : #000000;
	font-size : 110%;
}
P {
	background: #A9A9A9 none;
	color: #2F4F4F;
	font: 400 normal 120% Arial, Helvetica, sans-serif;
	text-decoration: none;
	padding: 2% 2% 2% 2%;
	border: 1px solid Black;
}
```

style2.css：

```
body {
	background-color: #FFFFF0;
}
a {
	background: #FFFFAA;
	text-decoration: none;
}
a:hover {
	background: #FFFFF0;
	border: thin solid Black;
}
h1 {
```

```
        text-align: center;
}
h2 {
        text-align: center;
}
p {
        text-align: center;
        font-size: 20px;
}
```

style3.css：

```
body {
        background-color: #FFFFF0;
        background-image: url(image001.jpg);
}
a {
        text-decoration: none;
        background-color: #0000CD;
        color: White;
        font-size: 15px;
}
a:hover {
        background-color: Maroon;
}
h1 {
        text-align: center;
        background-color: #FAFAD2;
        color: Blue;
}

h2 {
        text-align: center;
        background-color: #F5F5DC;
        color: Green;
        border: 1px solid Black;
}

p {
        text-align: center;
        border: 5px ridge;
        font-size: 40px;
}
```

由于 style3.css 使用了图片作为背景，所以要把名称为 image001.jpg 的图片放到该 css 文件的同一个目录下。一切准备就绪，先运行 11-8.php 文件，其结果如图 11.8 所示。由于此时并没有名为 style 的 Cookie 变量存在，所以现在不使用任何样式。

单击“风格 1”链接后，11-8.php 将注册 Cookie 值为 1 的变量 style。此时网页的效果将会如图 11.9 所示。

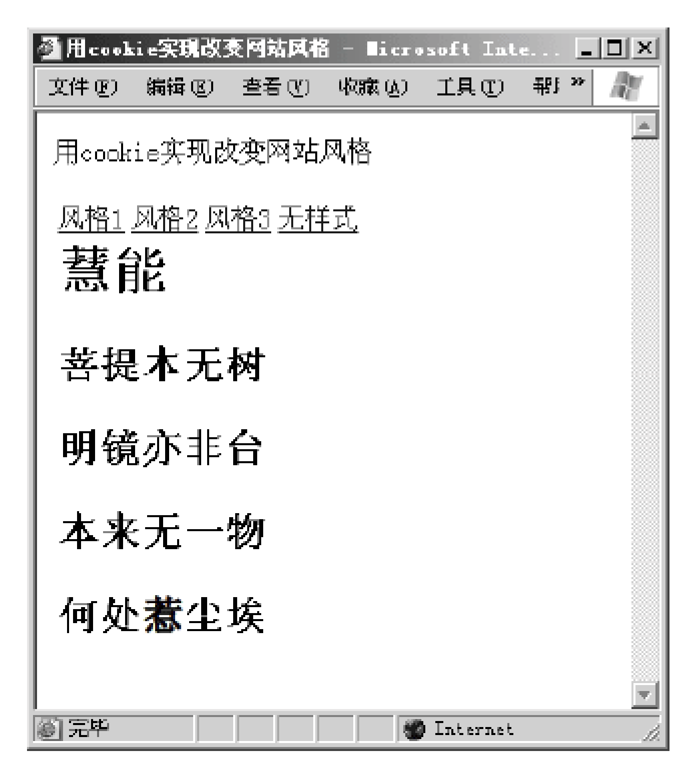


图 11.8　首次运行 11-8.php 文件时的效果

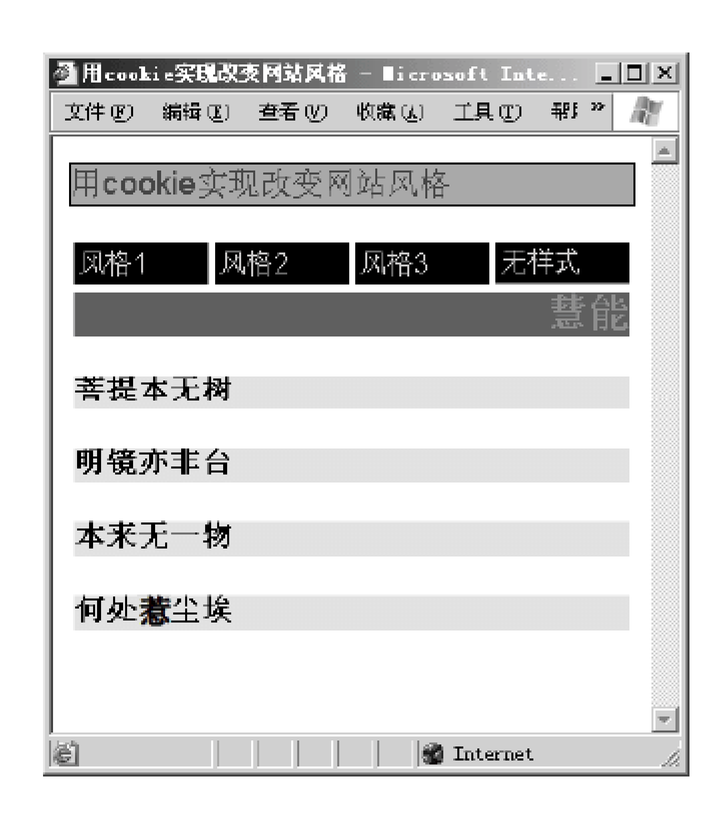


图 11.9　使用样式 1 的效果

由于此时已经有了名为 style 值为 1 的 Cookie 变量，所以此时使用样式 1。结果如图 11.9 所示。同理分别单击“风格 2”、“风格 3”链接，都会采用相应的样式文件。其执行结果分别如图 11.10、图 11.11 所示。

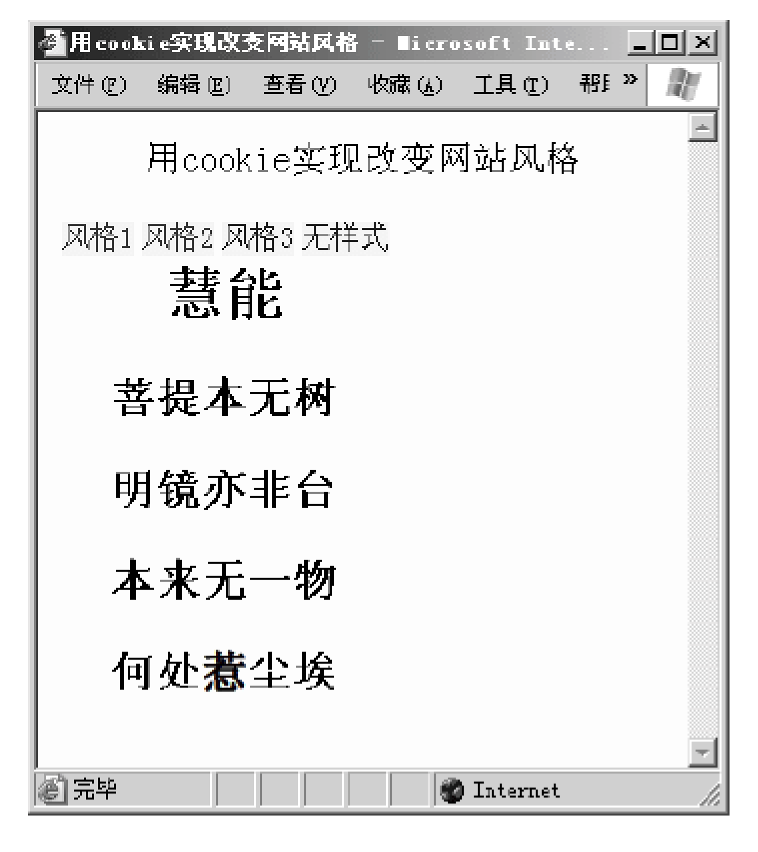


图 11.10　使用样式 2 的效果

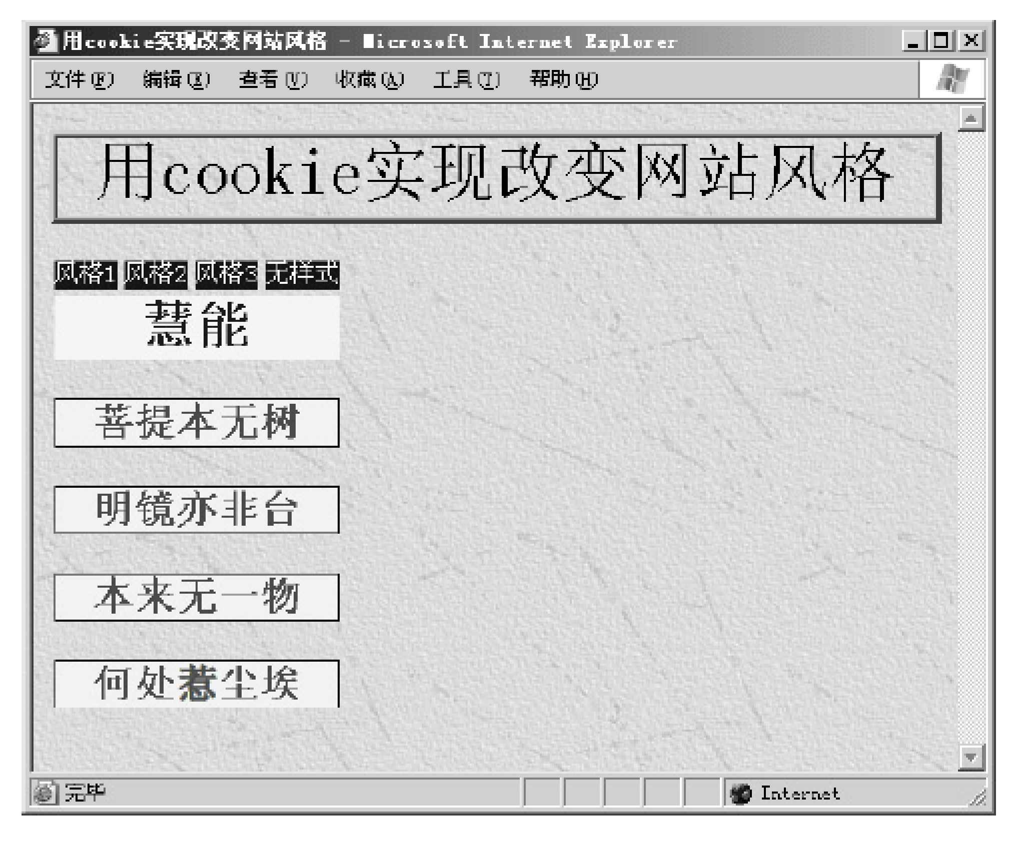


图 11.11　使用样式 3 的效果

如果单击“无样式”链接，由于没有参数 style，所以 11-9.php 中的 setcookie 一句的第 2 个参数为空，此时即为中止使用 Cookie 变量 style。即此时的 style 无效。所以返回的结果是同第一次打开时一样的效果。

至此使用 Cookie 实现网站风格转变的实例就为读者介绍完了。

11.3　本 章 小 结

这一章为读者介绍了 PHP 中的两种存储用户信息的机制，Session 和 Cookie。通过本章的学习，会使读者对如何使用这两种方式存储用户信息有一个比较深刻的认识。其实不管是采用 Session 作为存储载体还是采用 Cookie 作为存储的载体，关键要看怎么用。二者各有特点，有时，也可以把二者结合起来使用，取长补短。使自己用 PHP 编写的 Web 程序更稳定，更友好，才是学习的目的。

第12章

PHP 中正则表达式的使用

- 关于正则表达式
- POSIX 扩展的正则表达式函数
- Perl 兼容的正则表达式函数
- 正则表达式使用实例

正则表达式在 PHP 编程中有着相当广泛的应用，它可以实现对特定内容的查找、替换。本章将带领读者来学习 PHP 中的正则表达式。本章内容包括：什么是正则表达式、如何使用模式匹配、正则表达式语法、PHP 中的正则表达式函数、正则表达式使用实例等。通过本章的学习会使读者揭去正则表达式那神秘的面纱，一睹它的风采。

12.1　关于正则表达式

这一节将介绍关于正则表达式的基础性问题，让读者了解究竟什么是正则表达式、正则表达式由什么元素组成、如何使用正则表达式等。通过本节学习，将会使读者对正则表达式有一个比较全面的了解。

12.1.1　什么是正则表达式

简单的说，正则表达式就是一个字符构成的串，它定义了一个用来搜索匹配字符串的模式。它的作用是实现用正则表达式模式对一个字符串中特定的字符或字符集合进行查找与替换。正则表达式由一些普通字符和一些元字符（metacharacters）组成，它包含一个正则表达式模式。正则表达式中普通字符包括大小写的字母和数字，大多数字字符在模式中表示它们自身并匹配目标中相应的字符。而元字符则具有特殊的含义，后面会给予解释。

例如，判断邮政编码是否合法的正则表达式是像下面这样的内容：

```
ereg("^([0-9]{6})(-[0-9]{5})?$")
```

其中的“0-9”、“6”、“0-9”、“5”为普通字符，而里面的“^”、“[]”、“?”、“$”等为元字符。

正则表达式的优势在于其能够在模式中包含选择和循环。它们通过使用元字符来编码在模式中，元字符不代表其自身，它们用一些特殊的方式来解析。

有两组不同的元字符：一种是模式中除了方括号内的都能被识别，另一种是在方括号内的被识别。下面分别给读者作介绍。

方括号之外的元字符有以下内容，如表 12.1 所示。

表 12.1　正则表达式中的元字符列表

元　字　符	用在方括号内/外	作　　用
\	外	有数种用途的通用转义符
^	外	匹配字符串的开始（或在多行模式下行的开头，即紧随在换行符之后）
$	外	匹配字符串的结尾（或在多行模式下行的结尾，即紧随在换行符之前）
.	外	匹配除了换行符外的任意一个字符（默认情况下）
[	外	字符类定义开始
]	外	字符类定义结束
|	外	开始一个多选一的分支
(	外	子模式开始
)	外	子模式结束
?	外	扩展“(”的含义，也是0或1数量限定符，以及数量限定符最小值
*	外	匹配0个或多个的数量限定符
+	外	匹配1个或多个的数量限定符

续表

元　字　符	用在方括号内/外	作　　用
{	外	最少/最多数量限定开始
}	外	最少/最多数量限定结束
\	内	通用转义字符
^	内	排除字符类（逻辑非），但仅当其为第一个字符时有效
-	内	指出字符范围
]	内	结束字符类

表 12.1 中列出的模式在方括号内使用的部分称为“字符类”。

在以上介绍的几种元字符中，反斜线是一类特殊的字符，反斜线字符有 3 种用途。

第 1 个用途就是在它后面跟上非字母或数字的特殊符号来代替这些特殊符号本身。此种将反斜线用作转义字符的用法适用于以上两种情况。即不论是方括号之内还是方括号之外都适用。如要匹配一个“*”字符，则在模式中用“*”。这适用于无论下一个字符是否会被当作元字符来解释，因此在非字母数字字符之前加上一个“\”来指明该字符代表其本身总是安全的。尤其是，如果要匹配一个反斜线，用“\\”。

第 2 个作用是提供了一种在模式中以可见方式去编码不可打印字符的方法。并没有不可打印字符出现的限制，除了代表模式结束的二进制零以外。用文本编辑器来准备模式的时候，通常用以下的转义序列来表示那些二进制字符更容易一些，详见表 12.2。

表 12.2　反斜线通常用到的转义序列所表示的二进制字符列表

转 义 字 符	所表示的二进制字符
\a	alarm，即BEL字符（0x07）
\cx	“control-x”，其中x是任意字符
\e	escape（0x1B）
\f	换页符formfeed（0x0C）
\n	换行符newline（0x0A）
\r	回车符carriage return（0x0D）
\t	制表符tab（0x09）
\xhh	十六进制代码为hh的字符
\ddd	八进制代码为ddd的字符，或backreference
\040	另一种表示空格的方法
\40	同上，如果之前捕获的子模式少于40个的话
\7	总是一个逆向引用
\11	可能是个逆向引用，或者是制表符tab
\011	总是表示制表符tab
\0113	表示制表符tab后面跟着一个字符“3”
\113	表示八进制代码为113的字符（因为不能超过99个逆向引用）
\377	表示一个所有的比特都是1的字节
\81	要么是一个逆向引用，要么是一个二进制的零后面跟着两个字符“8”和“1”

注意八进制值的 100 或大于 100 的值之前不能以零打头，因为不会读取（反斜线后）超过 3 个八进制数字。

所有的定义了一个单一字节的序列可以用于字符类之中或之外。此外，在字符类之中，序列“\b”被解释为反斜线字符（0x08），而在字符类之外有不同含义（见下面）。

反斜线的第 3 个用法是指定通用字符类型。

- \d：任一十进制数字。
- \D：任一非十进制数的字符。
- \s：任一空白字符。
- \S：任一非空白字符。
- \w：任一“字”的字符。
- \W：任一“非字”的字符

反斜线的第 4 个用法是某些简单的断言。断言是指在一个匹配中的特定位置必须达到的条件，并不会消耗目标字符串中的任何字符。反斜线的断言如下。

- \b：字分界线。
- \B：非字分界线。
- \A：目标的开头（独立于多行模式）。
- \Z：目标的结尾或位于结尾的换行符前（独立于多行模式）。
- \z：目标的结尾（独立于多行模式）。
- \G：目标中的第 1 个匹配位置。

这些断言可能不能出现在字符类中（但是注意"\b"有不同的含义，在字符类之中也就是反斜线字符）。

12.1.2　如何使用模式匹配

12.1.1 小节为读者介绍了什么是正则表达式，及表达式中的元字符。这一小节就来介绍如何使用匹配模式。模式是正规表达式最基本的元素。它们是一组描述字符串特征的字符。模式可以很简单，由普通的字符串组成；也可以非常复杂，往往用特殊的字符表示一个范围内的字符重复出现，或表示上下文。

在最简单的情况下，一个正则表达式看上去就是一个普通的查找串。如正则表达式“testing”中没有包含任何元字符，它可以匹配“testing”和“123testing”等字符串，但是不能匹配“Testing”。下面详细为读者介绍如何使用模式匹配。

1．元字符“^”、“$”的使用

首先，介绍两个特别的元字符，“^”和“$”。上面说过它们是分别用来匹配字符串的开始和结束，以下分别举例说明。

“^The”

匹配以 The 开头的字符串。

“of despair$”

匹配以 of despair 结尾的字符串。

“^abc$”

匹配以 abc 开头和以 abc 结尾的字符串（实际上，只有 abc 与之匹配）。

“notice”

匹配包含 notice 的字符串。

从上面 3 个例子中可以了解到，如果没有用到上面的两个元字符（最后一个例子），就是说模式（正则表达式）可以出现在被检验字符串的任何地方，因为没有把它锁定到两边。

2．元字符“*”、“+”、“?”的使用

元字符“*”，“+”，和“?”用来表示一个字符可以出现的次数或者顺序。这 3 个字符分别表示：0 次或者多次；1 次或者多次；0 次或者 1 次。下面给出几个例子来说明这 3 个元字符是如何匹配字符的。

“ab*”

匹配字符串 a 和 0 个或者更多 b 组成的字符串。如“a”、“ab”、“abbb”等。

“ab+”

和上面一样，但最少有一个 b。如“ab”、“abbb”等。

“ab?”

匹配 0 个或者一个 b。只有两种可能“a”、“ab”。

3．综合使用“^”、“$”、“*”、“+”、“?”

下面把以上两种模式中的 5 个字符综合起来举几个例子。

“a?b+$”

匹配以一个或者 0 个 a 再加上一个以上的 b 结尾的字符串。如“b”、“ab”、“abb”、“abbb”等。

“^a*b?a$”

匹配以 0 个或者多个 a 开头再加上 0 个或者 1 个 b 再以 a 结尾的字符串。如“a”、“aa”、“abaaa”、“aabaa”等。

4．元字符“{”、“}”的使用

上面讲了“{”、“}”分别是数量限定符开始和结束的标记，所以也可以在大括号里面加上数字来限制字符出现的个数。如：

“ab{2}”

匹配一个 a 后面跟两个 b（一个也不能少）。如：“abb”。

“ab{2,}”

两个或者更多个“b”（即：最少更两个 b）。如“abb”、“abbbb”等。

“ab{3,5}”

a 后面跟 3~5 个 b（仅有 3、4、5 个 3 种情况）。如“abbb”、“abbbb”、“abbbbb”。

通过上面的例子，读者能够发现只能使用“{0,2}”，而不能使用“{,2}”。同样，“*”、“+”和“?”分别和以下 3 个范围标注是一样的，“{0,}”、“{1,}”和“{0,1}”。

5．元字符“(”、“)”的使用

介绍了大括号，再来介绍小括号。

现在把一定数量的字符放到小括号里。比如：

“a(bc)*”

匹配 a 后面跟 0 个或者 1 个“bc”。如“a”、“abc”（只有此两种情况）。

“a(bc){1,5}”

匹配 a 后面跟 1 个到 5 个“bc”。如“abc”、“abc”、“abcbc”、“abcbc”等。

6．元字符“|”的使用

还有一个字符“｜”，相当于 OR（或）操作。如：

“hi | hello”

匹配含有“hi”或者“hello”的字符串。

“(b | cd)ef”

匹配含有“bef”或者“cdef”的字符串。

“(a | b)*c”

匹配含有这样 0 到多个（包括 0 个）a 或 b，后面跟一个 c 的字符串。

7．元字符“.”的使用

下面介绍“.”符号。一个点('.')可以代表所有的单一字符。如：

“a.[0-9]”

匹配一个 a 跟一个字符再跟一个数字的字符串。

“^a.{3}$”

匹配以 a 开头 3 个任意字符结尾。

8．元字符“[”、“]”的使用

“[]”中括号括住的部分只匹配单一字符。

“[ab]”

匹配单个的 a 或者 b（和“a | b”一样）。

“[a-d]”

匹配“a”到“d”的单个字符（和“a | b | c | d”还有“[abcd]”效果一样）。

“^[a-zA-Z]”

匹配以字母开头的字符串。

“[0-9]%”

匹配含有形如“x%”的字符串。

“,[a-zA-Z0-9]$”

匹配以逗号在加一个数字或字母结尾的字符串。

9．中括号内的元字符“^”的使用

具体使用时也可以把不想要得字符列在中括号里，只需要在中括号里面使用“^”作为开头。即中括号里面的“^”符号表示逻辑非。

“%[^a-zA-Z]%”

匹配两个百分号中间含有一个非字母的字符串。如“%1%”、“%5%”等。

正则表达式中匹配模式的使用，这一节就介绍完了。12.2 节将为读者介绍 PHP 中的正则表达式相关函数。

12.2　POSIX 扩展的正则表达式函数

12.1 节为读者介绍了 PHP 中的匹配模式，但是光有模式是不能做事情的，必须得有函数与之一起工作才能起作用。所以，这一节就来介绍 PHP 中的正则表达式函数。PHP 支持两种模式的正则表达式函数，即 PHP 中有两类不同的正则表达式函数。一类是比较常用的 POSIX（Portable Operating System Interface 可移植操作系统接口）扩展的正则表达式函数，另一类是 Perl 兼容的正则表达式函数。两组函数功能大同小异，但函数名称及使用方法却不相同。下面分别为读者作讲解。

12.2.1　替换字符串

string ereg_replace(string pattern, string replacement, string string)。正则表达式替换函数。此函数在 string 中扫描与 pattern 匹配的部分，并将其替换为 replacement。返回替换后的字符串。如果没有可供替换的匹配项则会返回原字符串。

下面举两个简单的例子来说明此函数是如何使用的。

```
<html>
<head>
<title>ereg_replace()函数使用实例 1</title>
</head>
<body>
<?
$string1="abcacbcbaaab";                        //定义字符串变量
$string2=ereg_replace("a","b",$string1);        //对字符串进行替换操作
echo "string1 的内容为：";
echo $string1;                                  //输出原字符串内容
```

```
echo "<p>";
echo "经过 ereg_replace 处理（将 a 替换为 b）过的内容为：";
echo "<p>";
echo $string2;                                    //输出处理过之后的内容
?>
</body>
</html>
```

上面的例子只是使用了简单的查找替换，并没有用到正则表达式模式。只是通过该例子先来了解一下 ereg_replace()函数是如何起作用的。保存以上代码为“12-1.php”。在 PHP 执行环境中运行该 PHP 文件，其执行结果将会如图 12.1 所示。

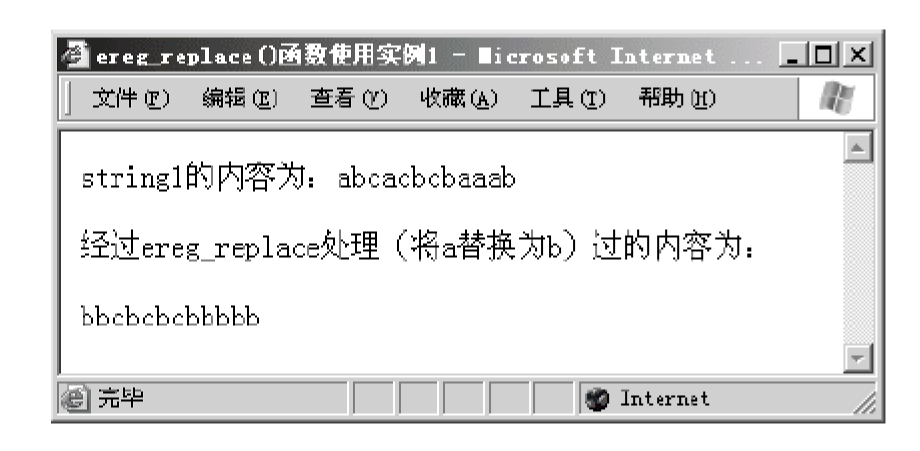


图 12.1　ereg_replace()函数使用实例 1 执行结果

通过上例及执行结果可见，经过 ereg_replace()函数的处理，字符串$string1 中所有为“a”的字符全被替换成了“b”。实现了替换字符串特定内容的目的。

下面再举一个稍微复杂一点的例子。这一次要替换的内容是包含有正则表达式模式的内容。具体内容参看以下代码：

```
<html>
<head>
<title>ereg_replace()函数使用实例 2</title>
</head>
<body>
<?
$string1="abcdacdbcbadaab";                       //定义字符串变量
$string2=ereg_replace("[a-b]","c",$string1);      //对字符串进行替换操作
echo "string1 的内容为：";
echo $string1;                                    //输出原字符串内容
echo "<p>";
echo "经过 ereg_replace 处理（将 a 和 b 替换为 c）过的内容为：";
echo "<p>";
echo $string2;                                    //输出处理过之后的内容
?>
</body>
</html>
```

保存以上代码为“12-2.php”文件。然后在 PHP 执行环境下执行该文件，执行结果如图 12.2 所示。

上面的例子就用到了正则表达式模式，实现的功能是把字符串$string1 中所有的 a 和 b 替换成 c。经过这样的处理，字符串中只有 c 和 d 了（注意观察$sting1 内容与上例的不同之处）。

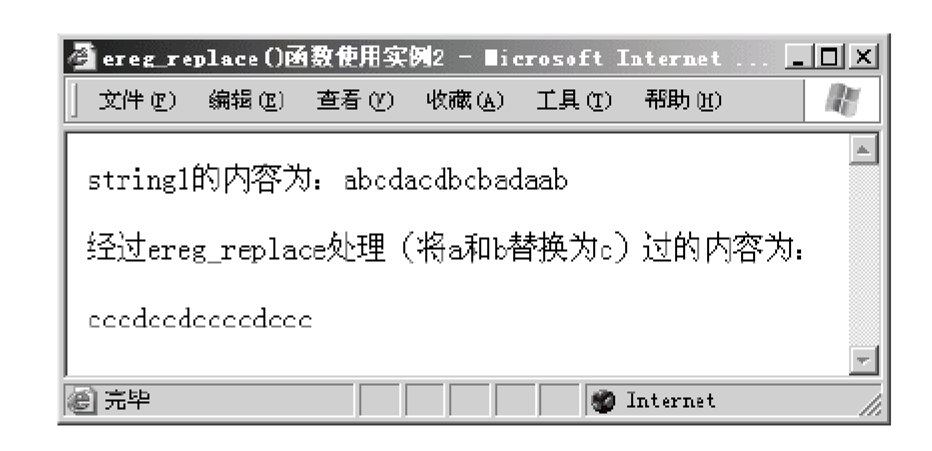


图 12.2　ereg_replace()函数使用实例 2 执行结果

12.2.2　匹配字符串

bool ereg(string pattern,string string[,array regs])函数对 string 字符串进行查找。如果在 string 中找到 pattern 模式的匹配，则返回 True，如果没有找到匹配或出错则返回 False。

注意此函数与 Perl 兼容正则表达式语法的 preg_match()函数具有相同的功能。通常使用 preg_match()函数是比 ereg()函数更快的替代方案。

下面举几个简单的例子来说明 ereg()函数是如何实现字符的匹配操作的。

```
<html>
<head>
<title>ereg()函数使用实例 1</title>
</head>
<body>
<?
$string1="abcde";                          //定义变量$string1
$string2="bbcde";                          //定义变量$string2
echo "string1 的内容为：";
echo $string1;                             //输出内容
echo "<p>";
echo "string2 的内容为：";
echo $string2;                             //输出内容
echo "<p>";
function panduan($s)                       //在 ereg()函数的基础上自定义一个函数
{
    if(ereg("a",$s)) echo "包含 a";         //根据 ereg()返回结果输出不同内容
    else echo "不包含 a";
}
echo "string1 中包含小写字母 a 吗？";
panduan($string1);                         //调用自定义函数进行判断
echo "<p>";
echo "string2 中包含小写字母 a 吗？";
panduan($string2);                         //调用自定义函数进行判断
?>
</body>
</html>
```

上面的例子中，只是进行了简单的判断，即在一个字符串中有没有包含另一个字符。其中并没有

用到正则表达式模式。保存以上代码为“12-3.php”。在 PHP 执行环境中运行该文件，执行结果如图 12.3 所示。

图 12.3　ereg()函数使用实例 1 执行结果

从上例及执行结果可以发现，通过在 ereg()函数基础上构建的自定义函数的判断，正确的输出了一个字符串中是否含有另一个字符（串）。下面再举一个例子，使用正则表达式模式对字符串进行是否匹配的判断。具体内容参看以下代码：

```
<html>
<head>
<title>ereg()函数使用实例 2</title>
</head>
<body>
<?
$string1="abcde";                                  //定义变量$string1
$string2="ABCD";                                   //定义变量$string2
echo "string1 的内容为：";
echo $string1;                                     //输出内容
echo "<p>";
echo "string2 的内容为：";
echo $string2;                                     //输出内容
echo "<p>";
function panduan($s)                               //在 ereg()函数的基础上自定义一个函数
{
     if(ereg("[a-z]",$s)) echo "包含小写字母";     //根据 ereg()返回结果输出不同内容
     else echo "不包含小写字母";
}
echo "string1 中包含小写字母吗？";
panduan($string1);                                 //调用自定义函数进行判断
echo "<p>";
echo "string2 中包含小写字母吗？";
panduan($string2);                                 //调用自定义函数进行判断
?>
</body>
</html>
```

保存以上代码为“12-4.php”。然后在 PHP 执行环境中运行该文件，其执行结果将会如图 12.4 所示。

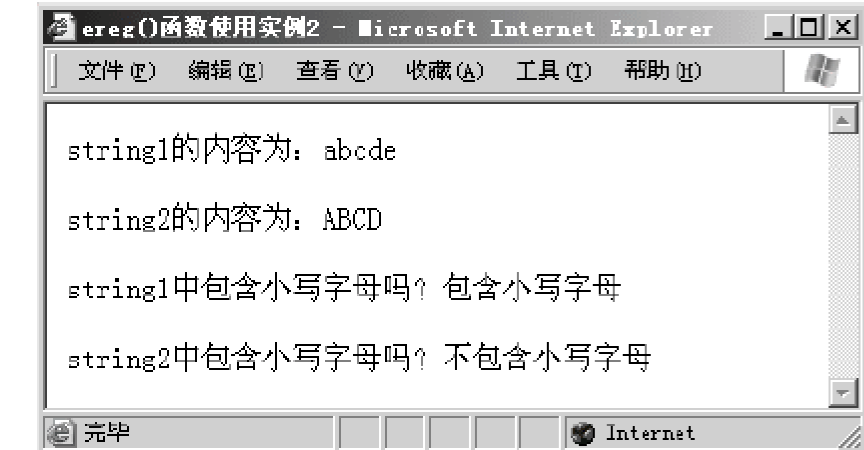


图 12.4　ereg()函数使用实例 2 执行结果

此次自定义的函数在使用 ereg()函数时，调用了正则表达式模式，即[a-z]（所有的小写字母）。对两个字符串中是否含有小写字母进行判断。从执行结果中也可以了解到，由于$string1 中包含了小写字母，所以 ereg()函数返回 True；而$string2 中没有包含小写字母（其中全为大写字母），ereg()函数返回了 False。通过该实例也可以发现此函数

是如何起作用的。

12.2.3　替换字符串（忽略大小写）

string eregi_replace(string pattern,string replacement,string string)。不区分大小写的正则表达式替换。此函数与 ereg_replace()函数有一样的功能，除了在替换字母字符时忽略字母的大小写。

下面通过一个例子来说明此函数与函数 ereg_repalce()的区别之处。具体内容参看以下代码：

```
<html>
<head>
<title>eregi_replace()函数使用实例</title>
</head>
<body>
<?
$string1="abcacbcbAAAb";                                    //定义字符串变量
$string2=eregi_replace("a","b",$string1);                   //对字符串进行替换操作
echo "string1 的内容为：";
echo $string1;                                              //输出原字符串内容
echo "<p>";
echo "经过 eregi_replace 处理（将 a 替换为 b）过的内容为：";
echo "<p>";
echo $string2;                                              //输出处理过之后的内容
?>
</body>
</html>
```

这个例子只是对 12-1.php 进行了简单的修改。其中的改动一个是把函数 ereg_replace()替换为函数 eregi_replace()，另一个也相应的把$string1 内容进行了调整。保存以上代码为“12-5.php”。在 PHP 执行环境下执行该文件，其执行结果将会如图 12.5 所示。

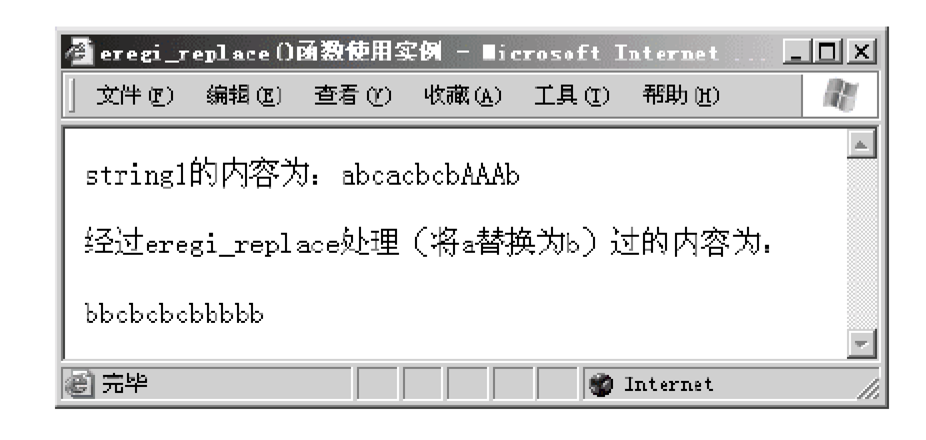


图 12.5　eregi_replace()函数使用实例执行结果

从图 12.5 中可以了解，由于忽略了字母字符的大小写，所以大写字母 A 也被替换成了小写字母 b。由于该函数与 ereg_replace()函数功能大致相似，所以就不再多举实例。有兴趣的读者可以把 12-2.php 文件作上述修改以查看二者的区别。

12.2.4　匹配字符串（忽略大小写）

bool eregi(string pattern,string string[,array regs])不区分大小写的正则表达式匹配。此函数与 ereg()函数也有一样的功能，不同的是在进行正则表达式的匹配字母字符时也忽略字母的大小写。

接下来仍然通过一个例子来说明该函数的使用。通过例子读者就会发现该函数与 ereg()函数的区别。

```
<html>
<head>
<title>eregi()函数使用实例</title>
</head>
<body>
<?
$string1="abcde";                              //定义变量$string1
$string2="ABCD";                               //定义变量$string2
echo "string1 的内容为：";
echo $string1;                                 //输出内容
echo "<p>";
echo "string2 的内容为：";
echo $string2;                                 //输出内容
echo "<p>";
function panduan($s)                           //在 eregi()函数的基础上自定义一个函数
{
    if(eregi("[a-z]",$s)) echo "包含小写字母";    //根据 eregi()返回结果输出不同内容
    else echo "不包含小写字母";
}
echo "string1 中包含小写字母吗？";
panduan($string1);                             //调用自定义函数进行判断
echo "<p>";
echo "string2 中包含小写字母吗？";
panduan($string2);                             //调用自定义函数进行判断
?>
</body>
</html>
```

对比以上代码与 12-4.php，可以发现此实例是对 12-4.php 进行的简单的修改，通过修改读者也可以发现两个函数在处理字母字符时的区别。保存以上代码为“12-6.php”。在 PHP 执行环境下运行该文件，其执行结果将会如图 12.6 所示。

查看图 12.6 执行结果，肯定会有些读者不理解，为什么只包含大写字母的字符串$string2 会被说成含有小写字母呢？因为这里使用了 eregi()函数。因为该函数在进行匹配时是忽略字母字符的大小写的，所以才会出现这样的情况。同时这也提醒了用户在进行大小写字母精确匹配时应该使用 ereg()函数，而不能使用 eregi()函数。

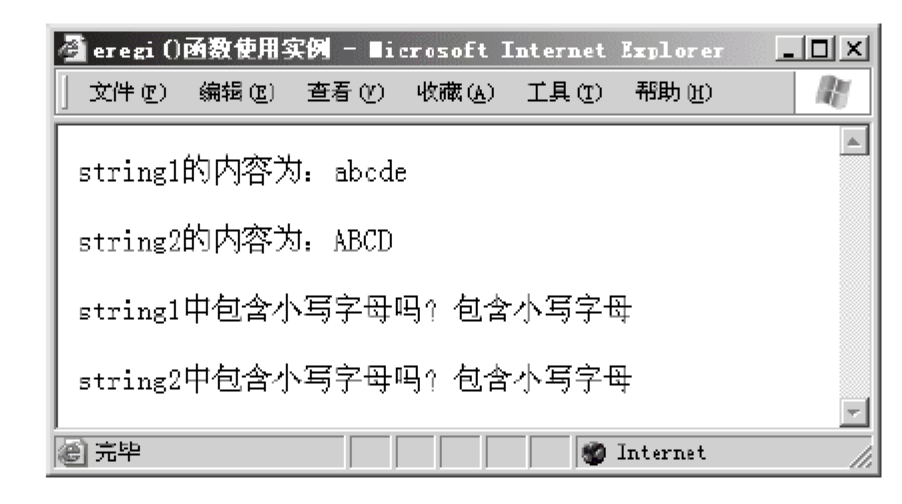


图 12.6　eregi()函数使用实例执行结果

12.2.5　分割字符串到数组

array split(string pattern,string string[,int limit])使用正则表达式将字符串分割到数组中。该函数返回一个字符串数组，数组的每个元素为 string 经区分大小写的正则表达式 pattern 作为边界分割出的子串。如果设定了 limit，则返回的数组最多包含 limit 个单元，而其中最后一个单元包含了 string 中剩余的所有部分。如果出错，则 split()返回 False。

另外，使用了 Perl 兼容正则表达式语法的 preg_split()函数，通常是比 split()函数更快的替代方案。如果不需要正则表达式的威力，则使用数组操作函数中的 explode()函数，会更快。这样就不会招致正则表达式引擎的浪费。

下面举两个简单的例子，使读者通过具体的实例来理解 split()函数是如何把字符串切分为数组的。

```
<html>
<head>
<title>split()函数使用实例 1</title>
</head>
<body>
<?
$string1="ab:cd:efgh:i";                          //定义变量$string1
echo "string1 的内容为：";
echo $string1;                                    //输出内容
echo "<p>";
echo "经过 split 处理过之后的数组内容为：";
echo "<p>";
$temp=split(":",$string1);                        //对字符串进行 split 处理
for($i=0;$i<count($temp);$i++)                    //通过循环输出数组内容
{
     echo "temp[".$i."]为：";
     echo $temp[$i];
     echo "，";
}
?>
</body>
</html>
```

保存以上代码为“12-7.php”。然后在 PHP 执行环境中执行该文件，执行结果如图 12.7 所示。

从图 12.7 可以了解到，经过 split()函数的处理，把原来的字符串变量$string1 按照“:”分割成了数组。从而实现了把字符串切分为数组的功能。

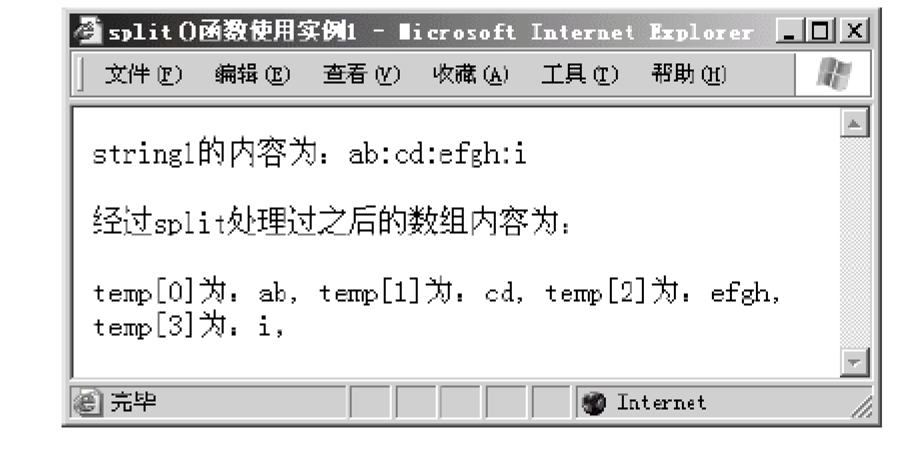


图 12.7　split()函数使用实例 1 执行结果

但是，12-7.php 只是实现了以普通字符作为分割标准。那么，如果以正则表达式模式作为分割标准又会怎么样呢？下面继续通过一个实例来说明，以正则表达式模式作为标准时，split()函数是如何起作用的。具体内容请参看以下代码：

```
<html>
<head>
<title>split()函数使用实例 2</title>
</head>
<body>
<?
$string1="ab123cd2efgh890i";                      //定义变量$string1
echo "string1 的内容为：";
```

```
echo $string1;                                //输出内容
echo "<p>";
echo "经过 split 处理过之后的数组内容为：";
echo "<p>";
$temp=split("[0-9]+",$string1);               //对字符串进行 split 处理
for($i=0;$i<count($temp);$i++)                //通过循环输出数组内容
{
    echo "temp[".$i."]为：";
    echo $temp[$i];
    echo "，";
}
?>
</body>
</html>
```

保存以上代码为“12-8.php”。然后在 PHP 执行环境下执行该文件，其执行结果将如图 12.8 所示。

从以上代码可以发现 12-8.php 与 12-7.php 的不同之处。12-7.php 在进行字符串割时所使用的分割标准是一个字符“;”，而 12-8.php 对字符串分割时使用的分割标准则是一个正则表达式模式“[0-9]+”，即 1 到多个数字，也就是用 1 到多个数字作为分割的标准。而输出结果也证明了结果的正确性。

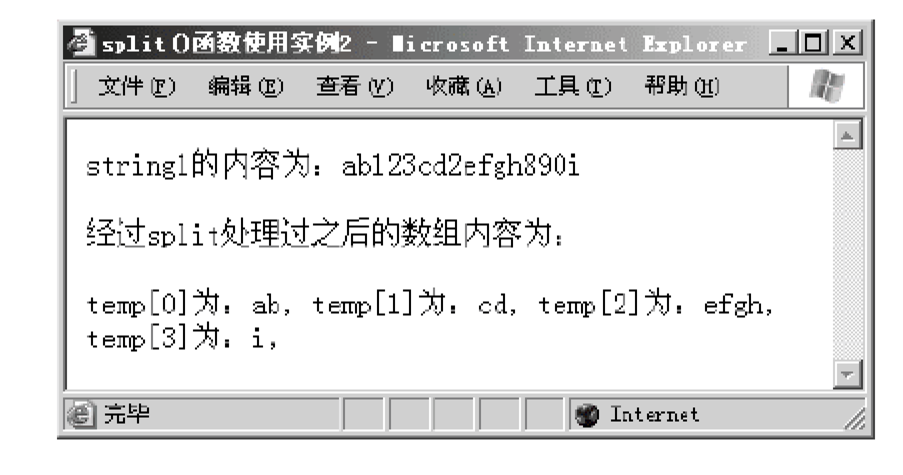


图 12.8　split()函数使用实例 2 执行结果

12.2.6　分割字符串到数组（忽略大小写）

array spliti(string pattern,string string[,int limit])函数实现功能与函数 split()完全一样，只是在对字符串进行分割时忽略字母字符的大小写。

由于此函数与 split()函数功能相似，唯一的不同就是在分割时忽略了字母字符的大小写。所以下面就只通过一个简单的例子来说明该函数是如何使用的。

```
<html>
<head>
<title>spliti()函数使用实例</title>
</head>
<body>
<?
$string1="abzcdZefghzi";                      //定义变量$string1
echo "string1 的内容为：";
echo $string1;                                //输出内容
echo "<p>";
echo "经过 spliti 处理过之后的数组内容为：";
echo "<p>";
$temp=spliti("z",$string1);                   //对字符串进行 split 处理
for($i=0;$i<count($temp);$i++)                //通过循环输出数组内容
{
```

```
        echo "temp[".$i."]为：";
        echo $temp[$i];
        echo "，";
    }
    ?>
    </body>
    </html>
```

保存以上代码为“12-9.php”，在 PHP 执行环境中执行该文件，执行结果将会如图 12.9 所示。

在 12-9.php 中，使用小写字母“z”来作为分割标准，但是字符串变量$string 中既含有小写字母“z”又有大写字母“Z”。如果是使用 split()函数进行分割，会只严格的使用小写字母，返回的数组只有 3 个元素。由于 spliti()函数是忽略字母字符，所以把大写字母也作为分割标准，这样一来，返回的数组就有 4 个元素了。

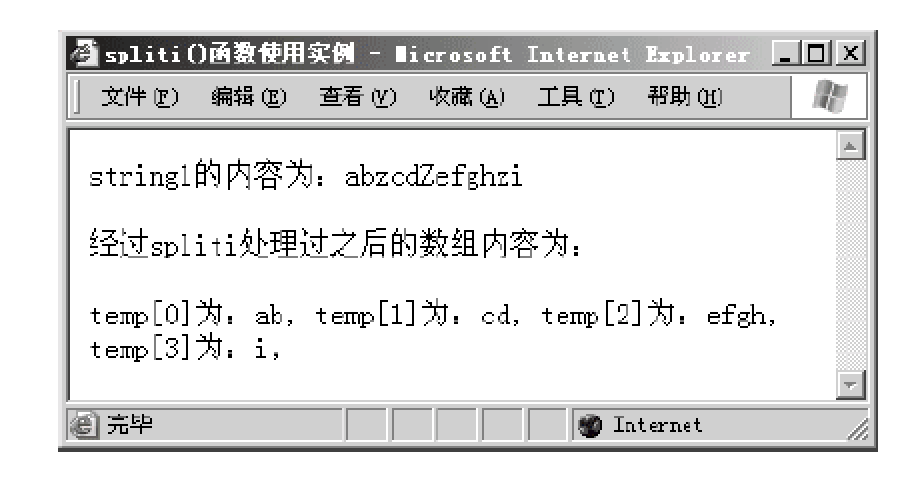


图 12.9　spliti()函数使用实例执行结果

12.2.7　返回包含指定字符的正则表达式

string sql_regcase(string string)。产生用于不区分大小写的匹配的正则表达式。此函数返回与 string 相匹配的正则表达式，忽略大小写字母。返回的表达式是将 string 中的每个字母字符转换为方括号表达式，该方括号表达式包含了该字母的大小两种形式。其他非字母字符保留不变。

下面举一个简单的例子来说明此函数是如何使用的。

```
<html>
<head>
<title>sql_regcase()函数使用实例</title>
</head>
<body>
<?
$string1="ab-zcd!Zefg?hzi";                 //定义变量$string1
echo "string1 的内容为：";
echo $string1;                              //输出内容
echo "<p>";
echo "经过 sql_regcase 处理过之后的内容为：";
echo "<p>";
$string2=sql_regcase($string1);             //对字符串进行 sql_regcase 处理
echo $string2;                              //输出处理过的结果
?>
</body>
</html>
```

保存以上代码为“12-10.php”，在 PHP 执行环境下执行该文件，其执行结果将会如图 12.10 所示。

从图 12.10 可以了解到，经过 sql_regcase 处理，原来的每个字母字符都转换为了方括号表达式，该方括号表达式包含了该字母的大小两种形式。该函数使用的场合就是对包含字母规则的正则表达式模式进行处理，使原本只支持大写或者小写的正则表达式模式（字串）转变成不区分大小写的正则表达式模式。

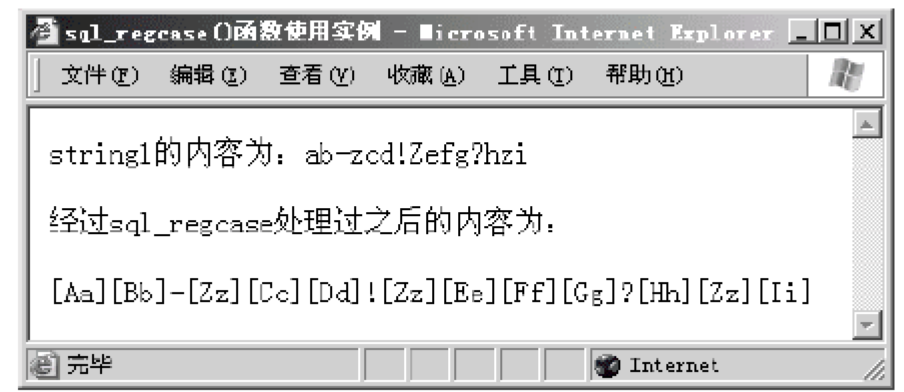


图 12.10　sql_regcase()函数使用实例执行结果

上面的几个就是使用 POXSI 扩展的正则表达式函数。下面来介绍使用 Perl 兼容模式的正则表达式函数。

12.3　Perl 兼容的正则表达式函数

PHP 有两组正则表达式函数。在 12.2 节介绍 POXSI 扩展的正则表达式函数的时候就曾提到有几个可以相互替代的函数。通常情况下，二者在某些函数功能上具有相似性，可以替代。

12.3.1　Perl 兼容正则表达式的使用规范

使用 Perl 兼容正则表达式函数的表达式应被包含在定界符中，如斜线（/）。任何不是字母、数字或反斜线（\）的字符都可以作为定界符。如果作为定界符的字符必须被用在表达式本身中，则需要用反斜线转义。自 PHP 4.0.4 起，也可以使用 Perl 风格的“()”，“{}”，“[]”和“<>”匹配定界符。结束定界符后可以跟上不同的修正符以影响匹配方式。

下面列出了当前在 PCRE 中可能使用的修正符。括号中是这些修正符的内部 PCRE 名。修正符中的空格和换行被忽略，其他字符会导致错误。

1．i (PCRE_CASELESS)

如果设定此修正符，模式中的字符将同时匹配大小写字母。

2．m（PCRE_MULTILINE）

默认情况下，PCRE 将目标字符串作为单一的一“行”字符所组成的（甚至其中包含有换行符也是如此）。“行起始”元字符（^）仅仅匹配字符串的起始，“行结束”元字符（$）仅仅匹配字符串的结束，或者最后一个字符是换行符时它的前面（除非设定了 D 修正符）。

当设定了此修正符，“行起始”和“行结束”除了匹配整个字符串开头和结束外，还分别匹配其中的换行符的之后和之前。这和 Perl 的/m 修正符是等效的。如果目标字符串中没有“\n”字符或者模式中没有^或$，则设定此修正符没有任何效果。

3．s（PCRE_DOTALL）

如果设定了此修正符，模式中的圆点元字符（.）匹配所有的字符，包括换行符。没有此设定的话，则不包括换行符。这和 Perl 的/s 修正符是等效的。排除字符类如[^a]总是匹配换行符的，无论是否设定了此修正符。

4. x（PCRE_EXTENDED）

如果设定了此修正符，模式中的空白字符除了被转义的或在字符类中的以外将完全被忽略，在未转义的字符类之外的#以及下一个换行符之间的所有字符，包括两头，也都被忽略。这和 Perl 的/x 修正符是等效的，使得可以在复杂的模式中加入注释。然而注意，这仅适用于数据字符。空白字符可能永远不会出现于模式中的特殊字符序列。如引入条件子模式的序列(?)中间。

5. e

如果设定了此修正符，preg_replace()在替换字符串中对逆向引用作正常的替换，将其作为 PHP 代码求值，并用其结果来替换所搜索的字符串。只有 preg_replace()使用此修正符，其他 PCRE 函数将忽略。

注意：此修正符在 PHP 3 中不可用。

6. A（PCRE_ANCHORED）

如果设定了此修正符，模式被强制为“anchored”，即强制仅从目标字符串的开头开始匹配。此效果也可以通过适当的模式本身来实现（在 Perl 中实现的唯一方法）。

7. D（PCRE_DOLLAR_ENDONLY）

如果设定了此修正符，模式中的美元元字符仅匹配目标字符串的结尾。没有此选项时，如果最后一个字符是换行符的话，美元符号也会匹配此字符之前（但不会匹配任何其他换行符之前）。如果设定了 m 修正符则忽略此选项。Perl 中没有与其等价的修正符。

8. S

当一个模式将被使用若干次时，为加速匹配起见值得先对其进行分析。如果设定了此修正符则会进行额外的分析。目前，分析一个模式仅对没有单一固定起始字符的 non-anchored 模式有用。

9. U（PCRE_UNGREEDY）

此修正符反转了匹配数量的值使其不是默认的重复，而变成在后面跟上“?”才变得重复。这和 Perl 不兼容。也可以通过在模式之中设定(?U)修正符或者在数量符之后跟一个问号（如.*?）来启用此选项。

10. X（PCRE_EXTRA）

此修正符启用了一个 PCRE 中与 Perl 不兼容的额外功能。模式中的任何反斜线后面跟上一个没有特殊意义的字母将导致一个错误，从而保留此组合以备将来扩充。默认情况下，和 Perl 一样，一个反斜线后面跟一个没有特殊意义的字母被当成该字母本身。当前没有其他特性受此修正符控制。

11. u（PCRE_UTF8）

此修正符启用了一个 PCRE 中与 Perl 不兼容的额外功能。模式字符串被当成 UTF-8。本修正符在 Unix 下自 PHP 4.1.0 起可用，在 win32 下自 PHP 4.2.3 起可用。自 PHP 4.3.5 起开始检查模式的 UTF-8 合法性。

12.3.2　返回匹配数组

array preg_grep(string pattern,array input)返回与模式匹配的数组元素。返回的数组包括了 input 数组

中与给定的 pattern 模式相匹配的元素。自 PHP 4.0.4 起，preg_grep()返回的结果使用从输入数组来的键名进行索引。如果不希望这样的结果，用 array_values()对 preg_grep()返回的结果重新索引。

下面通过一个例子说明此函数是如何从数组中提取出适合的模式匹配的。具体内容参看以下代码：

```
<html>
<head>
<title>preg_grep()函数使用实例</title>
</head>
<body>
<?
$temp[0]="abc";                                          //定义数组元素
$temp[1]=123;
$temp[2]="us.";
$temp[3]=5;
$temp[4]=58;
$temp[5]="CS";
echo "数组 temp 的内容为：";
echo "<P>";
for($i=0;$i<count($temp);$i++)                           //通过循环输出数组内容
{
     echo $temp[$i];
     echo "，";
}
echo "<p>";
echo "经过 preg_grep()处理过之后的新数组为：";
$temp2=preg_grep("/^(\d)\d*/",$temp);                    //使用 preg_grep 进行处理
//上式的正则表达式模式中“//”为定界符，其内容为所有整数元素
echo "<P>";
print_r($temp2);                                         //通过循环输出新数组内容
?>
</body>
</html>
```

保存以上代码为“12-11.php”，在 PHP 执行环境下执行该文件，其执行结果将会如图 12.11 所示。

从图 12.11 可以发现，经过 preg_grep()函数的处理，提取出了匹配 pattern 正则表达式的元素，组成了一个新的数组。然而新数组所使用的键名，依然是原来数组的键名。

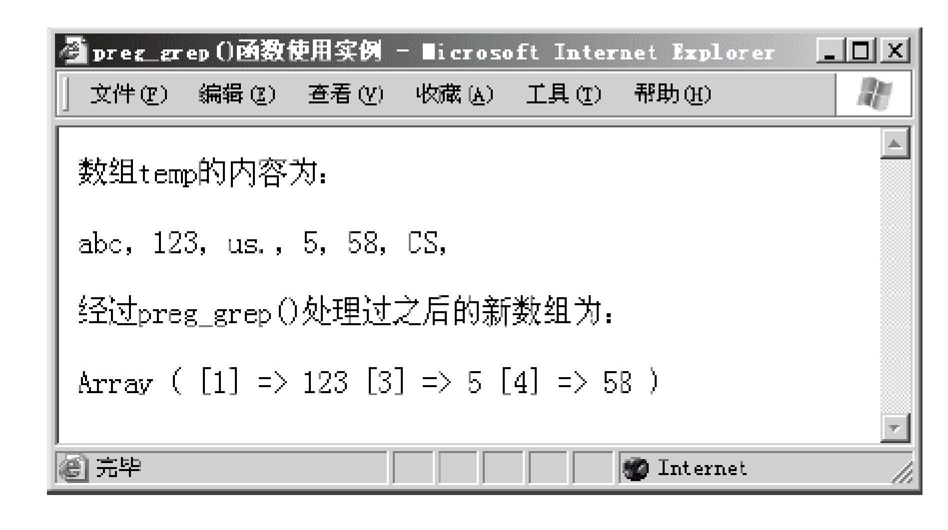


图 12.11　preg_grep()函数使用实例执行结果

12.3.3　全局表达式匹配

int preg_match_all(string pattern,string subject,array matches[,int flags])函数进行全局正则表达式匹配。在 subject 中，搜索所有与 pattern 给出的正则表达式匹配的内容并将结果以 flags 指定的顺序放到

matches 中。搜索到第 1 个匹配项之后，接下来的搜索从上一个匹配项末尾开始。函数返回整个模式匹配的次数（可能为零），如果出错返回 False。

flags 可以是下列标记的组合（注意把 PREG_PATTERN_ORDER 和 PREG_SET_ORDER 合起来用没有意义）：PREG_PATTERN_ORDER：对结果排序使$matches[0]为全部模式匹配的数组，$matches[1]为第 1 个括号中的子模式所匹配的字符串组成的数组，以此类推。

下面的一个例子就说明此函数是如何使用 PREG_PATTERN_ORDER 参数，并对结果进行排序的。具体内容参看以下代码：

```
<html>
<head>
<title>preg_match_all()函数使用实例 1</title>
</head>
<body>
<?
$string="<b>example: </b><div align=left>this is a test</div>";          //定义字符串
$pattern="|<[^>]+>(.*)</[^>]+>|U";                                       //定义正则表达式模式
preg_match_all($pattern,$string,$out,PREG_PATTERN_ORDER);                 //进行 preg_mathc_all 处理
echo $out[0][0];
echo "，";
echo $out[0][1];
echo "<p>";
echo $out[1][0];
echo "，";
echo $out[1][1];
?>
</body>
</html>
```

保存以上代码为“12-12.php”，在 PHP 执行环境下执行该文件，执行结果如图 12.12 所示。

先来分析$pattern 正则表达式模式所表达的含义。

```
$pattern="|<[^>]+>(.*)</[^>]+>|U";
```

其中的“|”、“|”为定界符，里面的内容表示什么含义呢？

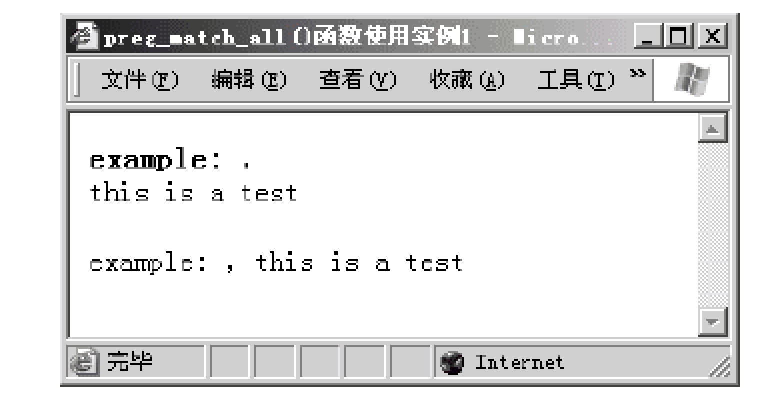


图 12.12　preg_match_all()函数使用实例 1 执行结果

第 1 个“<”表示以小于号开头的内容，后面跟 1 个到多个不为“>”的字符加上“>”结尾的内容。子模式中的“.*”表示 0 到多个任意字符。再加上以“<”开头的内容后面跟 1 个到多个不为“>”的字符加上“>”符号。模式修订符“U”的含义是反转了匹配数量的值使其不是默认的重复。

从图 12.12 可以发现，经过加上 PREG_PATTERN_ORDER 参数的 preg_match_all 函数的处理正确地返回了数组。

PREG_SET_ORDER 对结果排序。使$matches[0]为第 1 组匹配项的数组，$matches[1]为第 2 组匹配项的数组，以此类推。

下面通过一个的例子来说明函数是如何使用 PREG_SET_ORDER 对字符串进行提取及排序的。具

体内容参看以下代码：

```
<html>
<head>
<title>preg_match_all()函数使用实例 2</title>
</head>
<body>
<?
$string="<b>example: </b><div align=left>this is a test</div>";            //定义字符串
$pattern="|<[^>]+>(.*)</[^>]+>|U";                                          //定义正则表达式模式
preg_match_all($pattern,$string,$out,PREG_SET_ORDER);                      //进行 preg_match_all 处理
echo $out[0][0];
echo "，";
echo $out[0][1];
echo "<p>";
echo $out[1][0];
echo "，";
echo $out[1][1];
?>
</body>
</html>
```

这个例子是对上一个例子做了简单的修改，只是把其中的参数做了更换。保存以上代码为“12-13.php”，在 PHP 执行环境下执行该文件，其执行结果将会如图 12.13 所示。

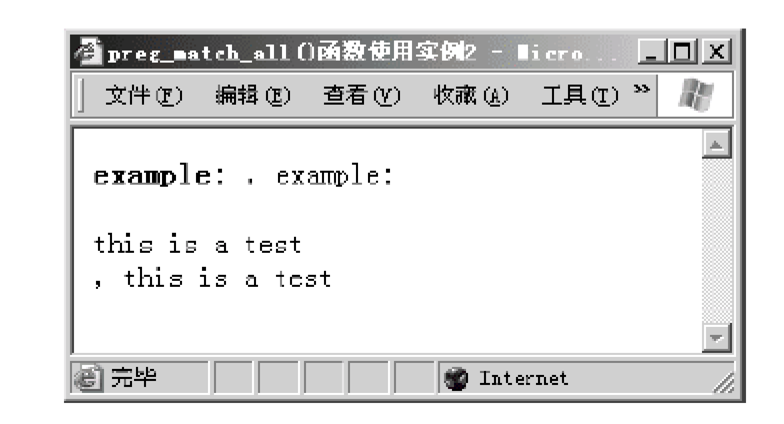


图 12.13　preg_match_all()函数使用实例 2 执行结果

从上例及图 12.13 的执行结果可以了解到，经过加上 PREG_SET_ORDER 参数的 preg_match_all 函数的处理使得$out[0]为第 1 组匹配的数组，$out[1]为第 2 组匹配的数组。从中也能了解使用此参数与 PREG_PATTERN_ORDER 参数的区别。

12.3.4　正则表达式匹配

int preg_match(string pattern,string subject[,array matches [,int flags]])函数进行正则表达式匹配。在 subject 字符串中，搜索与 pattern 给出的正则表达式相匹配的内容。如果提供了 matches，则其会被搜索的结果所填充。$matches[0]将包含与整个模式匹配的文本，$matches[1]将包含与第 1 个捕获的括号中的子模式所匹配的文本，以此类推。

这里通过实例来说明 preg_match()函数是如何实现对所给的字符串用正则表达式模式进行匹配的。

```
<html>
<head>
<title>preg_match()函数使用实例</title>
</head>
<body>
<?
```

```
$string="http://www.sohu.com";                          //定义字符串
$pattern="/^(http:\/\/)?([^\/]+)/i";                    //定义正则表达式模式
preg_match($pattern,$string,$matchs);                   //进行 preg_match 处理
echo "string 的内容为：";
echo $string;                                           //输出字符串变量内容
echo "<p>";
echo "经过 preg_match 函数处理过得出的主机名为：";
echo $matchs[2];                                        //输出取得数组的第 3 个元素
?>
</body>
</html>
```

以上正则表达式模式实现了从一个网址中取得主机名。如果执行无误，将会输出“www.sohu.com”。保存以上代码为“12-14.php”。然后在 PHP 执行环境下执行该文件，其执行结果将会如图 12.14 所示。

从图 12.14 所示的输出的执行结果中知道经过 preg_match()函数的处理（即使用正则表达式模式从一个网址中取得主机名），正确的得出了相匹配的结果。



图 12.14　preg_match()函数使用实例执行结果

12.3.5　转义正则表达式字符

string preg_quote (string str[,string delimiter])函数以 str 为参数并给其中每个属于正则表达式语法的字符前面加上一个反斜线。如果需要以动态生成的字符串作为模式去匹配，则可以用此函数转义其中可能包含的特殊字符。简单地说，此函数的目的就是使正则表达式中的特殊字符变成普通字符。

如果提供了可选参数 delimiter，该字符也将被转义。可以用来转义 PCRE 函数所需要的定界符，最常用的定界符是斜线“/”。正则表达式的特殊字符包括：“.”、“\\”、“+”、“*”、“?”、“[”、“^”、“]”、“$”、“(”、“)”、“{”、“}”、“=”、“!”、“<”、“>”、“|”。

下面为读者介绍两个简单的例子，通过例子来说明 preg_quote()函数是如何使正则表达式字符实现转义的。

```
<html>
<head>
<title>preg_quote()函数使用实例 1</title>
</head>
<body>
<?
$pattern="$40 for a g3/400";                            //定义正则表达式模式字符串
$out=preg_quote($pattern,"/");                          //进行 preg_mathc_all 处理
echo "pattern 的内容为：";
echo $pattern;                                          //输出字符串变量内容
echo "<p>";
echo "经过 preg_quote 函数处理过的结果为：";
```

```
echo $out;                                          //输出处理后的结果
?>
</body>
</html>
```

保存以上代码为“12-15.php”。在 PHP 执行环境中运行该文件，其执行结果将会如图 12.15 所示。

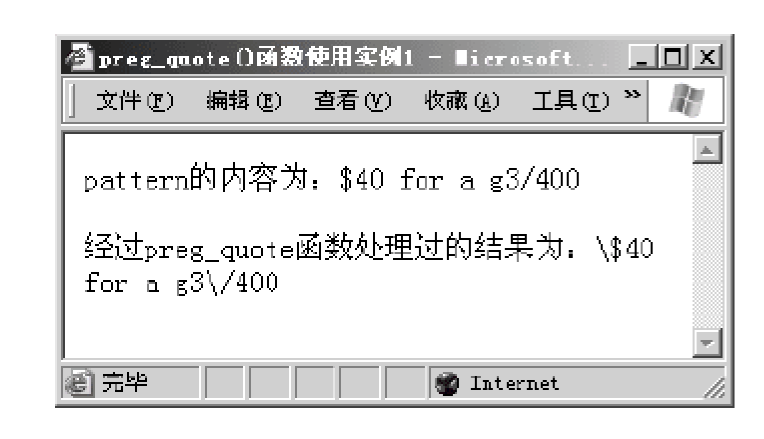


图 12.15　preg_quote()函数使用实例 1 执行结果

以下给出一个实例来说明，preg_quote()函数与 preg_replace()函数配合使用给某个单词加上斜线标记。具体内容参看以下代码：

```
<html>
<head>
<title>preg_quote()函数使用实例 2</title>
</head>
<body>
<?
$string="This book is *very* difficult to find.";          //定义字符串
$word="*very*";                                            //定义子字符串
$out=preg_quote($pattern,"/");                             //进行 preg_mathc_all 处理
echo "string 的内容为：";
echo $string;                                              //输出字符串变量内容
$stringout=preg_replace ("/".preg_quote($word)."/","<i>".$word."</i>",$string);
echo "<p>";
echo "经过 preg_quote 函数处理过的结果为：";
echo $stringout;                                           //输出处理后的结果
?>
</body>
</html>
```

保存以上代码为“12-16.php”，在 PHP 执行环境中执行该文件，其执行结果将会如图 12.16 所示。

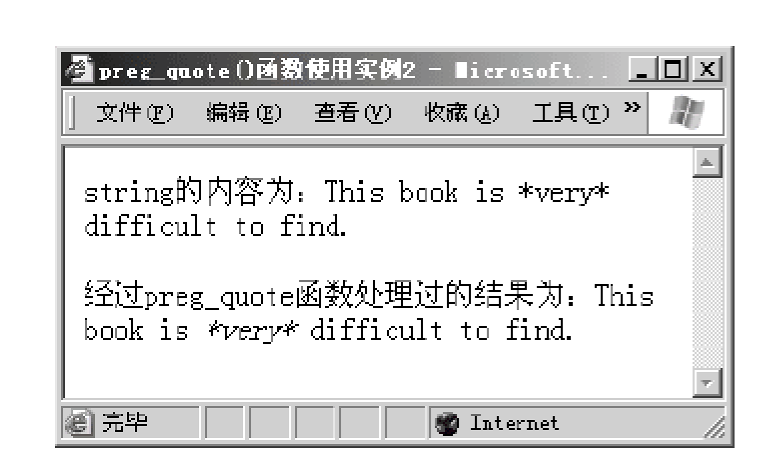


图 12.16　preg_quote()函数使用实例 2 执行结果

通过这两个例子的学习，对于 preg_quote()函数的使用，相信读者都会有一个比较清晰的认识了。

12.3.6　用回调函数实现正则表达式的搜索与替换

mixed preg_replace_callback(mixed pattern,callback callback,mixed subject[,int limit])可以执行正则表达式的搜索和替换。该函数的功能几乎和 preg_replace()一样，除了不是提供一个 replacement 参数，而是指定一个 callback 函数以外。该函数将以目标字符串中的匹配数组作为输入参数，并返回用于替换的字符串。

下面通过一个实例来说明 preg_repalce_callback()函数是如何使用回调函数执行正则表达式的搜索与替换的。

```
<html>
<head>
<title>preg_replace_callback()函数使用实例</title>
</head>
<body>
<?
$text="愚人节是：04/01/2005\n";
$text.="上一个圣诞节平安夜是：12/24/2004\n";                    //适应于 2005 年的文本内容
function next_year($matches)                                      //定义回调函数
{
//通常：$matches[0]是完整的匹配项
//$matches[1]是第一个括号中的子模式的匹配项
//以此类推
return $matches[1].($matches[2]+1);                               //返回函数的内容
}
echo $text;
echo "<p>";
echo preg_replace_callback("|(\d{2}/\d{2}/)(\d{4})|","next_year",$text);    //执行操作
?>
</body>
</html>
```

保存以上代码为“12-17.php”，在 PHP 执行环境中执行该文件，执行结果如图 12.17 所示。

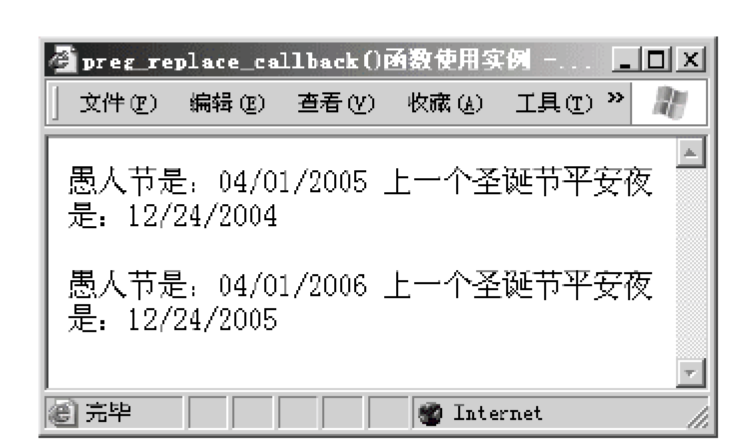


图 12.17　preg_replace_callback()函数使用实例执行结果

12.3.7　执行正则表达式的搜索与替换

mixed preg_replace(mixed pattern,mixed replacement,mixed subject[,int limit])。执行正则表达式的搜索和替换。在 subject 中搜索与 pattern 模式相匹配的项，并替换为 replacement。如果指定了 limit，则仅替换 limit 个匹配。如果省略 limit 或者其值为-1，则所有的匹配项都会被替换。

注意：limit 参数是 PHP 4.0.1 加入的。

接下来通过一个实例说明 preg_repalce()函数是如何进行搜索与替换的。下面函数实现的功能是去掉 HTML 文档内的 HTML 格式。具体内容参看以下代码：

```
<html>
<head>
<title>preg_replace ()函数使用实例</title>
</head>
<body>
<?
$search = array(
"'<script[^>]*?>.*?</script>'si",                    //去掉 javascript
"'<[\/\!]*?[^<>]*?>'si",                             //去掉 HTML 标记
"'([\r\n])[\s]+'",                                   //去掉空白字符
"'&(quot|#34);'i",                                   //替换 HTML 实体
"'&(amp|#38);'i",
"'&(lt|#60);'i",
"'&(gt|#62);'i",
"'&(nbsp|#160);'i",
"'&(iexcl|#161);'i",
"'&(cent|#162);'i",
"'&(pound|#163);'i",
"'&(copy|#169);'i",
"'&#(\d+);'e");                                      //作为 PHP 代码运行
$replace = array ("",
"",
"\\1",
"\"",
"&",
"<",
">",
" ",
chr(161),
chr(162),
chr(163),
chr(169),
"chr(\\1)");
$document="<b>I love this game</b><p><u><h1> 用 PHP 编 程 是 一 件 很 简 单 的 事
</h1></u><p><b><i><h2>HELLO WORLD！ </h2></i></b>";       //定义$doucment 变量
$text = preg_replace ($search, $replace, $document);      //使用 preg_replace 进行处理
echo "原来的内容为：";
echo $document;
echo "<p>";
echo "经过 preg_replace 处理之后的内容为：";
echo $text;
?>
</body>
</html>
```

把以上代码保存为“12-18.php”，在 PHP 执行环境中执行该文件，其执行结果会如图 12.18 所示。

从图 12.18 所示的执行结果可以发现，经过 preg_replace()的处理，原来包含 HTML 格式的内容，全部被转换成了普通字符输出。



图 12.18　preg_replace()函数使用实例执行结果

12.3.8　用正则表达式分割字符串

array preg_split(string pattern,string subject[,int limit[,int flags]])用正则表达式分割字符串。返回一个数组，包含 subject 中沿着与 pattern 匹配的边界所分割的子串。如果指定了 limit 参数，则最多返回 limit 个子串，如果没有指定 limit 或者其值是-1，则意味着没有限制，可以用来继续指定可选参数 flags。

flags 可以是下列标记的任意组合（用按位或运算符|组合）：PREG_SPLIT_NO_EMPTY，如果设定了本标记，则 preg_split()只返回非空的成分；PREG_SPLIT_DELIM_CAPTURE，如果设定了本标记，定界符模式中的括号表达式也会被捕获并返回，本标记添加于 PHP 4.0.5；PREG_SPLIT_OFFSET_CAPTURE，如果设定了本标记，对每个出现的匹配结果也同时返回其附属的字符串偏移量。注意这改变了返回的数组的值，使其中的每个单元也是一个数组，其中第 1 项为匹配字符串，第 2 项为其在 subject 中的偏移量。本标记从 PHP 4.3.0 起可用。

下面通过一个实例说明 preg_split()函数是如何对字符串进行分割的。具体内容参看以下代码：

```
<html>
<head>
<title>preg_split()函数使用实例</title>
</head>
<body>
<?
$str='string';
$chars=preg_split('//',$str,-1,PREG_SPLIT_NO_EMPTY);              //使用 PREG_SPLIT_NO_EMPTY 参数对
字符串进行分割，结果保存到数组$chars 中。
for($i=0;$i<count($chars);$i++)                                  //通过循环输出数组
{
      echo $chars[$i];
      echo "，";
}
?>
</body>
</html>
```

把以上代码保存为“12-19.php”，在 PHP 执行环境中执行该文件，其执行结果会如图 12.19 所示。

至此，PHP 中可以使用的两种正则表达式函数，POSIX 扩展模式、Perl 兼容模式的两组函数就全部介绍完毕。

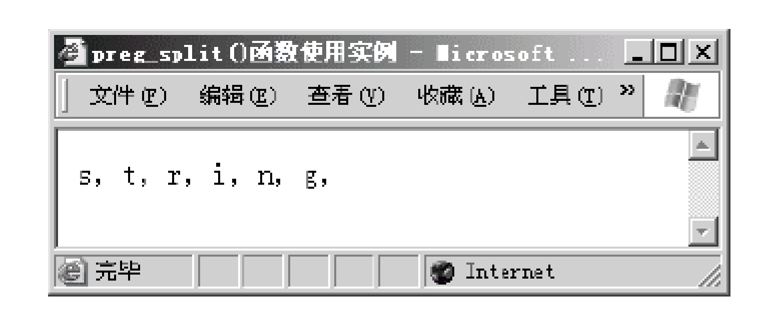


图 12.19　preg_split()函数使用实例执行结果

12.4　正则表达式使用实例

12.2，12.3 节为读者介绍了 PHP 中正则表达式里的元字符、表达式语法、正则表达式的两组函数。经过 12.2，12.3 节的学习，相信读者对在 PHP 中如何使用正则表达式都有一个比较清晰的认识了。这一节，将通过一些实际例子，来向读者做进一步的说明。这些例子都是在平常编程过程中经常可能遇到的功能。了解和使用这些例子对以后使用 PHP 编写 Web 程序将起到事半功倍的效果。

12.4.1　构造检查 E-mail 的正则表达式

这一小节来讨论怎么验证一个 E-mail 地址。通常一个完整的 E-mail 地址中有 3 个部分：POP3 用户名（在“@”左边的一切）、“@”、服务器名（就是剩下那部分）。用户名可以含有大小写字母阿拉伯数字、句号（“.”）、减号（“-”）以及 and 下划线（“_”）。服务器名字也是符合这个规则，当然下划线除外。

现在，用户名的开始和结束都不能是句点，服务器的名称也是这样。还有用户名中不能有两个连续的句点，它们之间至少应存在一个字符。下面就来说明怎么为用户名写一个匹配模式。代码如下：

```
^[_a-zA-Z0-9-]+$
```

现在的情况还不允许句号的存在。把它加上：

```
^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)*$
```

上面的正则表达式模式的意思是：“以至少一个规范字符（除.以外）开头，后面跟着 0 个或者多个以点开始的字符串”。

要使上面的正则表达式模式更简化一点，可以用 eregi()取代 ereg()。因为 eregi()在进行正则表达式模式匹配时忽略字母字符的大小写，这样就不需要指定两个范围：“a-z”和“A-Z”，只需要指定一个就可以了。改进之后的正则表达式模式如下：

```
^[_a-z0-9-]+(\.[_a-z0-9-]+)*$
```

后面的服务器名字也是一样，但要去掉下划线。代码如下：

```
^[a-z0-9-]+(\.[a-z0-9-]+)*$
```

设定好了用户名及主机名的规则，现在要做的只需要用“@”符号把用户名及主机名两部分内容连接起来。代码如下：

```
^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*$
```

这就是完整的 E-mail 认证匹配模式了，只需要调用以下代码：

```
eregi("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*$",$eamil)
```

就可以得出目标字符串是否为正确的 E-mail 地址了。

下面，通过一个实例来验证一下，以上的结果是否正确。

```
<html>
<head>
<title>使用正则表达式检查 email 地址</title>
</head>
<body>
<?
$string1="username@hostname.com";                    //定义变量$string1
$string2="user.name@host.com.cn";                    //定义变量$string2
$string3="@";                                        //定义变量$string3
$string4="username&hostname.com";                    //定义变量$string4
function panduan($string)                            //基于正则定义判断函数
{
    if(eregi("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*$",$string))    //判断
    echo $string."：是正确的 email 地址!";
    else
    echo $string."：不是 email 地址！";
}
panduan($string1);
echo "<p>";
panduan($string2);
echo "<p>";
panduan($string3);
echo "<p>";
panduan($string4);
?>
</body>
</html>
```

保存以上代码为“12-20.php”。然后在 PHP 执行环境下运行该文件，其执行结果将会如图 12.20 所示。



图 12.20　使用正则表达式检查 email 地址执行结果

12.4.2　对图像 UBB 代码进行替换

使用过网上的留言簿、论坛的用户都知道，一般情况下为了保证网络的安全是不允许用户直接使用 HTML 代码的，但是有的留言簿、论坛等 Web 程序却允许用户使用 UBB 代码。UBB 实际上就是把用户输入的特定内容转化为无害的 HTML 代码，这样既保证了安全，又可以使用户实现诸如贴图、发送移动文字、粘贴程序代码等功能。这一小节就带领读者来实现，如何把图 UBB 内容转化为带有图像的 HTML 代码。

UBB 图像代码的格式一般是这样：

```
[img]url[/img]
```

其中的 url 是指向一幅图片的地址。

列出了欲替换内容的格式，思路就很清晰了。匹配内容为：以“[img]”开头以“[/img]”结尾的内容，即上边的 URL 部分。

```
\[img\](.+)\[\/img\]
```

查找出内容之后就是替换了，把它替换成 HTML 代码：<img src=url>。这样就完成所要求的操作了。

调用 eregi_replace 函数：

```
eregi_replace("\[img\]","<img src=",$input);
eregi_replace("\[\/img\]",">",$input);
```

这样就完成了对 UBB 图像标记首尾的替换。这里为什么要调用 eregi_replace()函数，而不是 ereg_replace()函数呢？因为用户在输入的时候很有可能不会区分大小写，即大小写两种情况都有可能发生，所以要采用忽略大小写的 eregi_replace()函数。

下面通过一个实例来验证一下图片的替换是否能正常运行。

```
<html>
<head>
<title>对图像 UBB 代码进行替换</title>
</head>
<body>
<?
$string1="[img]1.jpg[/img]";                                    //定义变量$string1
$string2="[IMG]2.gif[/IMG]";                                    //定义变量$string2
$temp1=eregi_replace("\[img\]","<img src=",$string1);
$temp1=eregi_replace("\[\/img\]",">",$temp1);
$temp2=eregi_replace("\[img\]","<img src=",$string2);
$temp2=eregi_replace("\[\/img\]",">",$temp2);
echo "<table border=1>";
echo "<tr>";
echo "<td>string1 的内容为：<p>".$string1."</td>";
echo "<td>string2 的内容为：<p>".$string2."</td>";
echo "</tr>";
echo "<tr>";
echo "<td>将 string1 转化后的内容为：<p>".$temp1."</td>";
echo "<td>将 string2 转化后的内容为：<p>".$temp2."</td>";
echo "</tr>";
echo "</table>";
?>
</body>
</html>
```

保存以上代码为“12-21.php”，在 PHP 执行环境下运行该文件（在执行该文件之前，要把名为 1.jpg 及 2.gif 的两个图像文件放到该 PHP 文件同一路径下），其执行结果将会如图 12.21 所示。

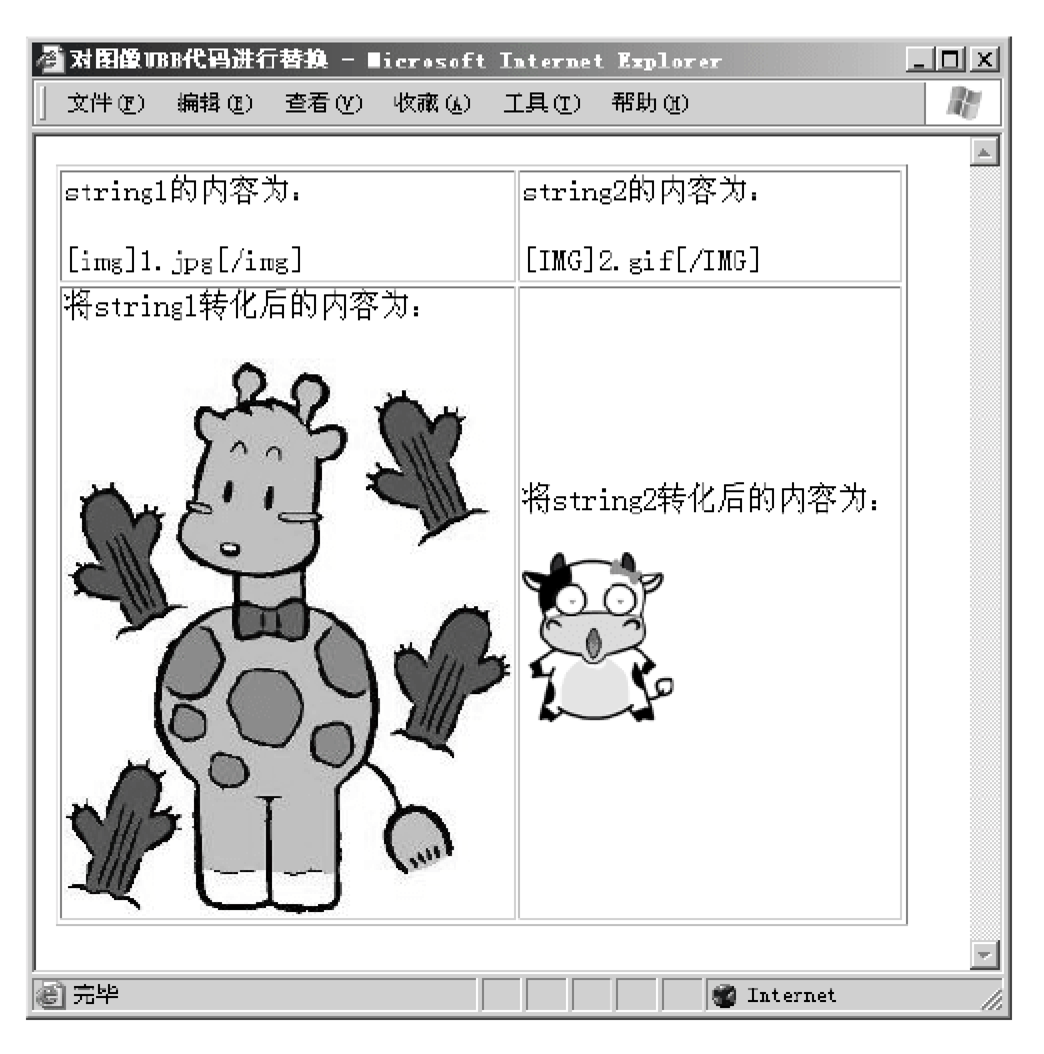


图 12.21　对图像 UBB 代码进行替换执行结果

从图 12.21 可以发现，经过对图像的替换操作，把原本为 UBB 代码的内容转化成了 HTML 内容，在 Web 页面上显示 HTML 内容，就显示出了图片。从而说明图像替换正则表达式的执行结果是正确的。

除了常见的查找、替换功能之外，正则表达式还具有如下两个功能。

❑　提取字符串

ereg()函数和 eregi()函数有一个特性是允许用户通过正则表达式去提取字符串的一部分。如：想从 path/URL 中提取文件名就可以执行如下面的操作。

```
ereg("([^\\/]*)$",$pathOrUrl,$regs);
echo $regs[1];
```

❑　高级的代换

ereg_replace()和 eregi_replace()也是非常有用的，假如用户想把所有的间隔负号都替换成逗号，就可以使用这样的操作。

```
ereg_replace("[ \n\r\t]+", ",", trim($str));
```

12.5　本 章 小 结

本章为读者介绍了 PHP 中的正则表达式，带领读者由简入繁地学习了什么是正则表达式、正则表达式的构成、正则表达式的语法、PHP 中可以使用的关于正则表达式的两组函数，以及通过具体的实例进一步巩固了正则表达式的使用。正如本章开头所说的那样，正则表达式在进行 PHP 编程中有着相当广泛的应用。通过本章的学习相信读者已经能够熟练地掌握正则表达式知识了，这将会使以后的 Web 编程工作变得更加轻松。

第13章 PHP面向对象编程

- 什么是面向对象编程
- 在PHP中创建和使用类
- PHP中与类相关的函数
- PHP中使用类的实例

通常的编程语言所使用的方法有两种：面向过程方法，如C语言所采用的方法；面向对象（OOP）方法，如Java所采用的方法。PHP是一种混合语言，它同时兼有面向过程和面向对象这两种能力。具体使用哪种模式，完全取决于用户的选择。两种模式各有其优劣。这一节将重点介绍PHP中的面向对象编程（OOP）。

本章内容包括：面向对象编程的基础、什么是类、为什么要用到类；在PHP程序中如何使用对象、类的封装、为类添加属性、为类添加方法、类的继承、类的重载、类的引用、类的串行化以及PHP常用的对象操作函数等内容。通过对本章的学习，将会使读者对在PHP中使用对象有一个全面的认识。

13.1　面向对象编程（OOP）基础

要想使用面向对象编程（Object Oriented Program），首先要了解什么是面向对象、面向对象的构成以及什么是类、为什么要用到类的问题。

13.1.1　什么是类（class）

在了解什么是类之前，先来了解一下面向对象到底是怎么一回事。在前几章为读者所提供的例子中，使用的代码基本上全部都是面对过程的。面向过程的代码一般看起来如下：

```
<?
echo "HELLO WORLD!";
?>
```

这就是通常所用到的方法。但是达到同一个目的，面向对象的代码却如下：

```
<?
class helloworld                              //定义一个类
{
      function myprint()                      //该类的一个方法
      {
            print "HELLO, WORLD.";            //该类的方法所实现的功能是打印字符串
      }
}
$myhelloworld =new helloworld();              //为类初始化一个实例（对象）
$myhelloworld->myprint();                     //调用该实例的方法
?>
```

上面这段代码就是标准的面向对象的做法。面向对象的做法通常是这样的：为了完成一件事，就要先定义一个类，然后给该类添加相应的方法，然后再为这个类派生一个对象，让对象去执行相应的方法，从而完成所需工作。

上例中，helloworld 是一个类，通过在类中添加了一个现在可以看作是一个函数的方法 myprint()，该方法完成的工作就是输出一个字符串“HELLO WORLD”。而$myhelloworld 则是类 helloworld 的一个对象。

从以上两组代码中，读者应该能够看得出面向过程与面向对象的区别。面向过程的基础是一句一句的代码，而面向对象的基础则是对象。

下面就来介绍什么是类。从简单的概念入手，什么是类？抛开各种教材的定义，通俗一点来说，“类”就是“一类某某事物”的简称，就这么简单。什么可以作为“类”呢？“人”就是一个类，与这个类相区别，“动物”和“植物”就是另外的两个类。类是具有相同的属性定义和行为表现的事物的集合。以“人”为例，这个类具有“国籍”、“种族”、“年龄”、“姓名”等属性定义，也具有“走路”、“工作”等行为表现。

但是，类不是指个体。我们可以说人具有国籍，但不能说“人”这个类的“国籍”是什么，因为“国籍”这个属性的表现是要随着类的具体化而实现。也就是说，需要具体到类中的一个单个的元素，也就是这个类的一个“实例”。类是虚无的东西，一个概念名词而已。真正用到的是类的实例，即对象。例如一个公司，它的雇员是一个一个的具体的人，而不是整个“人类”。套用某些教材里的话，类是现实世界某些对象的共同特征（属性和操作）的表示，对象是类的实例。

总而言之，类是变量（类的属性）与作用于这些变量的函数（类的方法）的集合。类包括该类的属性与方法，属性与方法是构成类的基础。

13.1.2　为什么要用到类

面向过程编程存在一个弊端，代码重用时非常困难。这在大规模编程中表现得尤其突出。面向过程编程时，如果用大量的代码来完成一个工作，当需要重用时，唯一的方法就是重新调用这些代码。任务量非常大，而且还很容易出错。

面向过程的这个弊端在面向对象编程模式中得到了很好的解决。类对代码进行了封装，所以在进行重用时就显得非常方便，只需要调用相关类的方法或属性就可以了。

所以使用面向对象编程具有如下优点：

- 易维护。采用面向对象思想设计的结构，可读性高。由于继承的存在，即使改变需求，维护也只是在局部模块，所以维护起来是非常方便和较低成本的。
- 质量高。在设计时，可重用现有的在以前的项目领域中已被测试过的类使系统满足业务需求并具有较高的质量。
- 效率高。在软件开发时，根据设计的需要对现实世界的事物进行抽象，产生类。使用这样的方法解决问题，接近于日常生活和自然的思考方式，势必提高软件开发的效率和质量。
- 易扩展。由于继承、封装、多态的特性，自然可以设计出高内聚、低耦合的系统结构，使得系统更灵活、更容易扩展，而且成本较低。

由于与面向过程的方法相比具有以上优点，所以用 PHP 在进行大型项目的开发时使用面向对象的编程方法是必要的。

13.2　在 PHP 中创建和使用类

以前在介绍变量时就曾提到过，变量有几种类型，其中有两种类型比较特殊，一种是数组，另一种是对象。在 PHP 中，对象可以看成是一种特殊的变量，对象有属性、方法。PHP 中具体怎么创建并使用类呢？这一节就来回答这个问题。

13.2.1　自建类——类的封装

前面提到过，类是变量与作用于这些变量的函数的集合。所以要创建类就离不开变量与函数。在介绍如何自建类之前，先来看一个实例。下面代码给出了如何创建一个类，并为类实例化一个对象及

如何调用类的方法。

1．创建类

在使用一个类之前必须要先创建类。创建类的语法结构如下：

```
class classname
{
}
```

创建类的方法由一个 class 关键字再加一个类的名称构成。以下实例构建一个购物车 cart 类。代码如下：

```
<?
class cart
{
	var $items;					//购物车中的物品
	function add_item($artnr,$num)			//将$num 个$artnr 物品加入购物车
	{
		$this->items[$artnr]+=$num;		//现有数量加上放入数量
	}
	function remove_item($artnr,$num)		//将$num 个$artnr 物品从购物车中取出
	{
		if($this->items[$artnr]>$num)		//如果取出数量小于现有总数
		{
			$this->items[$artnr]-=$num;	//现有数量减去取出数量
			return true;			//返回真
		}
		elseif($this->items[$artnr]==$num)	//如果要取出数量与现有数量相等
		{
			unset($this->items[$artnr]);	//清空现有数量值
			return true;
		}
		else					//如果取出数比现有数还要大
		{
			return false;			//返回错误
		}
	}
}
```

上面的代码创建了一个名为 cart 的类。这个类由购物车中的商品构成的数组和两个用于从购物车中添加和删除商品的函数组成。接下来再来讲解如何为类实例化一个对象。

2．创建对象

只创建了类，而不为类实例化一个对象，那么类就不会起作用。创建一个类的目的就是要为了使用类来实例化对象。下面就来介绍如何为类实例化对象。

创建对象的语法如下：

```
$objectname=new classname
```

即对象名（记得前面的美元符号）加“=”符号再加上“new”加上类名即可。

上面的例子就创建了$mycart 这样一个基于类 cart 的对象。

```
$mycart=new cart;                              //创建$mycart 对象
$mycart->add_item("5",3);                      //调用$mycart 对象的放入购物车方法
```

上例中的$mycart 具有方法 add_item()，remove_item()和一个 items 变量。它们都是明显的函数和变量。在引用对象的变量与函数时使用$mycart->items 及$mycart->additems()。所以调用一个对象的变量与函数使用下面的语法格式：

```
$objectname->varname;
$objectname->functionname;
```

在使用一个对象的时候有一个问题需要注意，如上例中的变量名为$cart->items，而不是$cart->$items，那是因为在 PHP 中一个变量名只有一个单独的美元符号。

另外，在定义类的时候，无法得知将有什么名字的对象来访问。如在编写 cart 类时，并不知道后面对象的名称将会命名为$mycart 或者其他的名称，因而不能在类中使用$cart->$items。然而为了类定义的内部访问自身的函数和变量，可以使用伪变量$this 来达到这个目的。$this 变量可以理解为“我自己的”或者“当前对象”，因而'$this->items[$artnr]+=$num'可以理解为“我自己的物品数组的$artnr 计数器加$num”或者“在当前对象的物品数组的$artnr 计数器加$num”。

下面把前面的例子代码加以完善，让其有实际的执行结果，从而使读者直观地认识到如何定义类，实例化对象，引用类的变量及函数。代码如下：

```
<html>
<head>
<title>定义与使用类实例</title>
</head>
<body>
<?
class cart
{
    var $items;                                  //购物车中的物品
    function add_item($artnr,$num)               //将$num 个$artnr 物品加入购物车
    {
        $this->items[$artnr]+=$num;              //现有数量加上放入数量
    }
    function remove_item($artnr,$num)            //将$num 个$artnr 物品从购物车中取出
    {
        if($this->items[$artnr]>$num)            //如果取出数量小于现有总数
        {
            $this->items[$artnr]-=$num;          //现有数量减去取出数量
            return true;                         //返回真
        }
        elseif($this->items[$artnr]==$num)       //如果要取出数量与现有数量相等
        {
            unset($this->items[$artnr]);         //清空现有数量值
            return true;
```

```
            }
            else                                        //如果取出数比现有数还要大
            {
                return false;                           //返回错误
            }
        }
    }
    $mycart=new cart;                                   //创建$mycart 对象
    $mycart->add_item("5",3);                           //调用$mycart 对象的放入购物车方法
    echo "在实现 add_item(\"5\",3)之后";
    echo "<p>";
    echo "当前购物车内编号为 5 的商品数量为：";
    echo $mycart->items["5"];                           //显示对象的 items["5"]变量
    echo "<p>";
    $mycart->add_item("5",10);                          //调用$mycart 对象的 add_item()函数
    echo "在实现 add_item(\"5\",10)之后";
    echo "<p>";
    echo "当前购物车内编号为 5 的商品数量为：";
    echo $mycart->items["5"];                           //显示对象的 items["5"]变量
    echo "<p>";
    $mycart->remove_item("5",7);                        //调用$mycart 对象的 remove_item()函数
    echo "在实现 remove_item(\"5\",7)之后";
    echo "<p>";
    echo "当前购物车内编号为 5 的商品数量为：";
    echo $mycart->items["5"];                           //显示对象的 items["5"]变量
    ?>
    </body>
    </html>
```

保存以上代码为“13-1.php”，然后在 PHP 执行环境下运行该文件，将会出现如图 13.1 所示的执行结果。

从图 13.1 可以看出，在实现 add_item("5",3)后（即向购物车内添加 3 个编号为 5 的商品），编号为 5 的商品现在在购物车内的数量为 3；实现 add_item("5",10)后(即向购物车内添加 10 个编号为 5 的商品），数量再增加 10 变为 13；实现 remove_item("5",7)（即把购物车内个编号为 5 的商品减去 7 个）后，数量减去 7 变为了 6。说明类通过调用类的方法，改变了对象的某些属性。



图 13.1　定义与使用类实例执行结果

13.2.2　为类添加属性

使用 class 语句创建类后，为类添加必要的属性（即变量）是必要的，也是必须的。通常情况下，类都有属性。

现在回过头再来看 13-1.php。其中，cart 是一个类，而里面的 items 就是类的一个属性。类的属性

通俗地说也就是存在于类中的变量。

知道了这样的概念，就明白如何为类添加属性了。为类添加属性就是向类中加入新的变量。下面通过一个实例来具体说明如何为类添加属性。代码如下：

```
<html>
<head>
<title>为类添加属性</title>
</head>
<body>
<?
class User                                  //首先定义一个类
{
      var $name;                            //用户的名字
      var $age;                             //用户的年龄
      var $sex;                             //用户的性别
}
$user1=new User;                            //创建一个对象
$user1->name="Jack";                        //为名字赋值
$user1->age="20";                           //为年龄赋值
$user1->sex="male";                         //为性别赋值
echo "user1 的名字为：";
echo $user1->name;                          //显示名字
echo "<p>";
echo "user1 的年龄为：";
echo $user1->age;                           //显示年龄
echo "<p>";
echo "user1 的性别为：";
echo $user1->sex;                           //显示性别
?>
</body>
</html>
```

把以上代码保存为“13-2.php”，然后在 PHP 执行环境下运行该文件，其执行结果如图 13.2 所示。

上例中，首先创建了一个类 User，然后为类添加了 3 个属性，分别为用户的名称、用户的年龄、用户的性别。然后为类实例化了一个对象$user1，又给对象的每个属性赋予了值。调用对象的属性显示出相应的内容。

注意：在类中，为类添加属性所定义的变量，只能使用关键字 var 来定义。



图 13.2　为类添加属性执行结果

为类添加属性就介绍到这里。13.2.3 小节介绍如何为类添加方法。

13.2.3　为类添加方法

方法就是类中的函数。调用类的方法就是执行其中的函数。所以为类添加方法，也就是给类中增

加函数。

仍然通过 13-1.php 来研究如何为类添加方法。在 13-1.php 中，cart 是一个类，其中有两个函数，即 add_item()和 remove_item()，这两个函数也就是类 cart 的方法。而类的方法也就是存在于类中的函数。

要实现为类中添加方法，只需向类中加入新的函数就可以了。下面通过实例来说明如何为类添加方法。代码如下：

```
<html>
<head>
<title>为类添加方法</title>
</head>
<body>
<?
class Window                                    //首先定义一个类
{
      var $state;                               //窗户的状态
      function close_window()                   //关窗户方法
      {
            $this->state="close";               //窗户的状态为关
            return true;
      }
      function open_window()                    //开窗户方法
      {
            $this->state="open";                //窗户的状态为开
            return true;
      }
}
$mywindow=new Window;                           //创建一个对象
$mywindow->state="open";                        //窗户的初始状态为开
echo "当前 mywindw 的状态为：";
echo $mywindow->state;                          //显示窗户的状态
echo "<p>";
$mywindow->close_window();                      //调用关窗户的方法
echo "经过 close_windw()处理后的的状态为：";
echo $mywindow->state;                          //显示窗户的状态
echo "<p>";
$mywindow->open_window();                       //调用开窗户的方法
echo "经过 open_windw()处理后的的状态为：";
echo $mywindow->state;                          //显示窗户的状态
?>
</body>
</html>
```

保存以上代码为“13-3.php”。然后在 PHP 执行环境中运行该文件，其执行结果将会如图 13.3 所示。

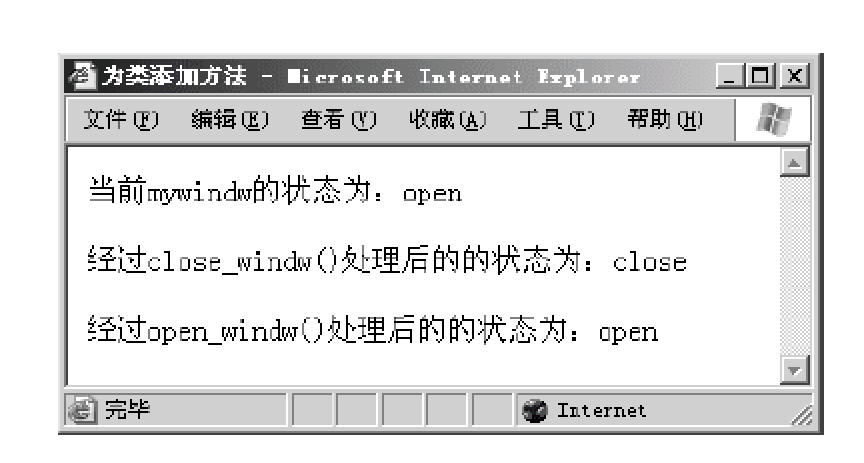


图 13.3　为类添加方法执行结果

上例中，定义了一个 window 类，并为类添加了$state 属性，即窗户的开关状态。然后为类添加

了两个方法，一个是 close_window()关窗户的方法，另一个是 open_window()开窗户的方法。使用这两个方法，可以改变窗户的开关状态。定义完类后，为类创建了一个对象。可以看出，通出两个方法是如何改变窗户的开关状态的。

在类中为类添加方法和平常定义一个函数并没有本质上的区别。关键是如果函数中要引用类本身的状态或者函数要使用伪变量$this 加上想要引用的变量或者函数名。这一点需要引起读者的注意。

13.2.4　类的继承

这一小节来研究类的继承问题。何为类的继承呢？通常需要这样一些类，这些类与其他现有的类拥有相同的变量和函数。为了使这一点变得更加容易，类可以从其他类中扩展出来。扩展或派生出来的类拥有其基类或者父类的所有变量和函数，并包含所有派生类中定义的部分。类中的元素不可能减少，也就是说不可以注销任何存在的函数或者变量。一个扩充类总是依赖一个单独的基类，也就是说不支持多重继承。

在 PHP 中使用如下语句来实现对一个类的继承：

```
class Son_class extends class
```

就是在定义一个子类 Son_class 时后面加上 extends 再加上父类的名称 class 即可。

下面把 13-3.php 的代码做一下修改，以便讲解如何从一个父类中继承子类。如下：

```
<html>
<head>
<title>类的继承使用实例</title>
</head>
<body>
<?
class Window                                    //首先定义一个类
{
        var $state;                             //窗户的状态
        function close_window()                 //关窗户方法
        {
                $this->state="close";           //窗户的状态为关
        }
        function open_window()                  //开窗户方法
        {
                $this->state="open";            //窗户的状态为开
        }
}
class Who_Window extends Window                 //定义一个子类继承父类
{
        var $owner;                             //加入属性窗户的所有人
}
$my_who_window=new Who_window;                  //实例化一个对象
$my_who_window->owner="Jack";                   //定义窗户的所有人
echo "my_who_window 的 owner 为：";
echo "<p>";
```

```
echo $my_who_window->owner;                      //显示窗户的所有人
echo "<p>";
$my_who_window->close_window();                  //调用关窗户事件
echo "my_who_window 的开关状态为：";
echo "<p>";
echo $my_who_window->state;                      //显示窗户开关状态
?>
</body>
</html>
```

把以上代码保存为“13-4.php”，然后在 PHP 执行环境下执行该文件，执行结果将会如图 13.4 所示。

下面来分析一下 13-4.php。首先定义了一个父类 window，该类有一个 state 属性与开关窗户两个方法。然后又定义了一个子类 who_window，由于使用了 extends 关键字，并且后面跟上了 window 的类名，所以该子类是继承自父类 window。所以，who_window 类就拥有父类的全部属性与方法，并且该子类也有自己的属性$owner，即窗户的所有人。



图 13.4　类的继承使用实例执行结果

为类 who_window 实例化了一个对象 my_who_window。由于该对象是基于 who_window 创建的，而 who_window 又是继承于父类 window。那么该对象就既有$state 属性也有$owner 属性，同时也有开关窗户的两种方法。而后面的执行结果也证实了这一点。

13.2.5　类的重载

一个子类中的属性或方法有时会与它所继承的父类中的属性或方法重名，这时就出现了类的重载。

类的重载实际上是类属性以及类的方法的重载。那么什么是类的方法的重载呢？通过前面的介绍了解到，定义一个父类后，可以再定义一个子类对其继承。这时就存在一个问题，如果子类中又定义了和父类中一样的变量和函数，会出现什么后果呢？这时，如果执行子类的某个方法，还将执行子类中同名的方法，而并不会影响父类中相应的方法。这一过程就叫做类的重载。

当然在子类中也可以访问父类中的方法。不过要使用关键字 self 和 parent，这两个特殊的关键字是用于在类的内部对成员或方法进行访问的。

下面使用一个实例来向读者说明如何使用类的重载，以及如何在子类中对其父类的方法的访问。代码如下：

```
<html>
<head>
<title>类的重载使用实例</title>
</head>
<body>
<?
class Window                                     //首先定义一个类
{
```

```
        var $state;                                    //窗户的状态
        function close_window()                        //关窗户方法
        {
            $this->state="close";                      //窗户的状态为关
        }
        function open_window()                         //开窗户方法
        {
            $this->state="open";                       //窗户的状态为开
        }
    }
    class Who_Window extends Window                    //创建一个子类
    {
        var $owner;                                    //加入属性
        function close_window()                        //方法的重载
        {
            $this->state="close";                      //此处改变窗户的状态
            $this->owner="Jack";                       //此处改变窗户的所有者
        }
        function open_window()                         //方法重载
        {
            parent::open_window();                     //执行父类的方法，并不改变所有者
        }
    }
    $my_who_window=new Who_window;                     //创建一个基于子类的对象
    $my_who_window->close_window();                    //调用关窗户事件
    echo "my_who_window 的状态为：";
    echo "<p>";
    echo $my_who_window->state;                        //窗户状态此时为关
    echo "<p>";
    echo "my_who_window 的 owner 为：";
    echo "<p>";
    echo $my_who_window->owner;                        //窗户的所有者现在为 jack
    echo "<p>";
    $my_who_window->owner="Li";                        //改变窗户所有者
    $my_who_window->open_window();                     //调用开窗户事件
    echo "my_who_window 的状态为：";
    echo "<p>";
    echo $my_who_window->state;                        //显示状态
    echo "<p>";
    echo "my_who_window 的 owner 为：";                //显示所有者
    echo "<p>";
    echo $my_who_window->owner;
    ?>
    </body>
    </html>
```

把以上代码保存为“13-5.php”，然后在 PHP 执行环境下运行该文件，其执行结果将会如图 13.5 所示。

上例中的子类中含有和父类中同名的方法，即进行了重载。其中的 close_window()方法，不仅把窗户开关状态设置为关，而且还把窗户的主人设置为 Jack。而 open_window()方法只是执行父类中的方法改变窗户的开关状态，并没有改变窗户的主人。从这里也可以看到，在子类中引用父类方法的使用模式。要使用 parent 关键字，后面还要加上“::”双冒号符号，然后在后面加上方法名。例如下面的语法：

```
parent::method_name()
```

从以上例子及执行结果中可以了解到方法的重载是如何使用的。

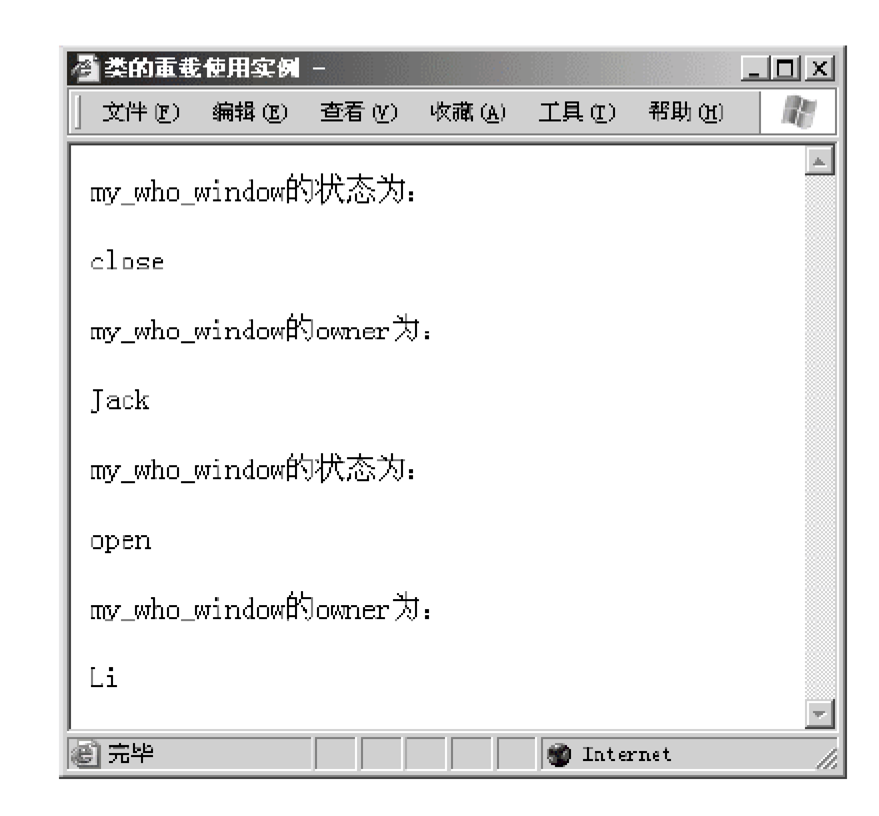


图 13.5　类的重载使用实例执行结果

13.2.6　类的引用

在一个类中使用了其他的类的属性或方法就称为类的引用。虽然这在理论上是行的，但不推荐这么做。因为使用类的目的就是把一类有共同属性的东西封装起来，如果在这些有共同属性的东西中再引用别的东西，感觉很不协调，这也不符合面向对象编程的初衷。不过有关联的类与类之间还是可以互相引用的，关键看这些类的关系。

下面通过一个实例来说明如何实现类的引用。代码如下：

```
<html>
<head>
<title>类的引用使用实例</title>
</head>
<body>
<?
class A                                         //创建类 A
{
        function a_method()                     //创建方法
        {
                echo "I'm classA's method!";    //输出相应内容
        }
}
class B                                         //创建类 B
{
        function b_method()                     //创建方法
        {
                echo "I'm classB's method!";    //输出相应内容
                A::A_method();                  //引用类 A 中的方法
        }
}
$my_b=new B;                                    //为类 B 实例化一个对象
$my_b->b_method();                              //调用对象的方法
```

```
?>
</body>
</html>
```

把以上代码保存为“13-6.php”，然后在 PHP 执行环境下运行该文件，其执行结果将会如图 13.6 所示。



图 13.6　类的引用使用实例执行结果

通过以上实例及其执行结果可以看出，在一个类中是可以引用另外一个类的方法的。使用如下语法：

```
class_name::method_name;
```

即类名加上双冒号再加上方法名。可以发现，这与在子类中引用父类的方法有相似之处。那个不需要使用类名，而只用 parent 关键字就代替了父类的类名。

13.2.7　类的构造函数

构造函数是类中的一个特殊函数，当使用 new 操作符创建一个类的实例时，构造函数将会自动调用。PHP 3 中，当函数与类同名时，这个函数将成为构造函数。PHP 4 中，在类中定义的函数与类同名时，这个函数将成为一个构造函数。

可以这样简单地理解，构造函数就是当对象被创建时类中一定发生作用的函数。下面来看看在 PHP 中如何创建并使用构造函数。代码如下：

```
<html>
<head>
<title>类的构造函数使用实例</title>
</head>
<body>
<?
class Window                                        //首先定义一个类
{
	function Window()
	{
		$this->close_window();
	}
	var $state;                                     //窗户的状态
	function close_window()                         //关窗户方法
	{
		$this->state="close";                       //窗户的状态为关
	}
	function open_window()                          //开窗户方法
	{
		$this->state="open";                        //窗户的状态为开
	}
}
$my_window=new Window;
echo "my_window 的开关状为：";
```

```
echo "<p>";
echo $my_window->state;
?>
</body>
</html>
```

保存以上代码为“13-7.php”，并在 PHP 执行环境下运行该文件，其执行结果将会如图 13.7 所示。

为什么窗户的状态会是关的？因为在创建对象时启动了类的构造函数。在上例中，构造函数 Window()的内容是什么呢？是调用本类的 close_window()方法。由于创建对象时启动了构造函数 Window()，而 Window()函数又是调用了 close_window()方法。也就是说，在创建对象时就把窗户的开关状态设置为关了，所以才会出现如图 13.7 所示的执行结果。

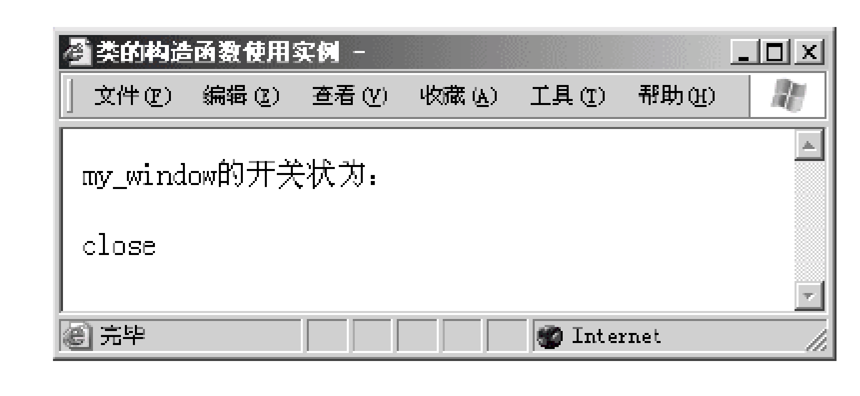


图 13.7　类的构造函数使用实例执行结果

还有一点，如果构造函数带有参数，那么在为类创建对象时也需要带上参数。如果没有参数就会有“缺少参数”的出错提示。下面通过一个简单的例子来说明这个问题。代码如下：

```
<html>
<head>
<title>构造函数的参数使用实例</title>
</head>
<body>
<?
class Temp                                    //首先定义一个类
{
        function Temp($str)
        {
                echo $str;
        }
}
$my_temp=new Temp("hello world!");
?>
</body>
</html>
```

在 PHP 执行环境中运行该文件，将会输出“hello world!”的字样。这是因为类 Temp 的构造函数使用了参数。所以创建对象时也必须带上参数。如果去掉参数，则会出现出错提示。

在使用构造函数时要注意一个问题是，当创建一个对象时，如果类没有构造函数，而父类有构造函数，则父类的构造函数将会被调用。

这个问题也很容易理解，这里就不再赘述。

13.3　PHP 中与类、对象相关的函数

PHP 中提供了函数来实现对类的相关的操作。本节为读者介绍 PHP 中与类相关的常用函数。

- bool class_exists(string class_name[,bool autoload])函数：检查类是否已定义。如果由字符串变量 class_name 所指的类已经定义，则此函数返回 True，否则返回 False。

在进行 PHP 编程中，判断一个类是否被定义也是很必要的，如果类已经被定义就可以为该类创建一个对象。但如果类不存在，也就无法创建对象了。

下面通过一个实例来说明如何使用 class_exists()函数来判断类是否被定义。具体内容参见以下代码：

```
<html>
<head>
<title>class_exists()函数使用实例</title>
</head>
<body>
<?
class Window                                    //首先定义一个类
{
        var $state;                             //窗户的状态
        function close_window()                 //关窗户方法
        {
                $this->state="close";           //窗户的状态为关
        }
        function open_window()                  //开窗户方法
        {
                $this->state="open";            //窗户的状态为开
        }
}
class Who_Window extends Window                 //创建子类
{
        var $owner;
        function close_window()                 //方法继承
        {
                $this->state="close";
                $this->owner="Jack";
        }
}
function f_e($string)                           //创建一个基于 class_exists()的自定义函数
{
        if(class_exists($string))               //如果类名存在
        {
                echo "名为".$string."的类已经存在！";    //打印相应信息
        }
        else                                    //如果类不存在
        {
                echo "名为".$string."的类并不存在！";    //打印相应信息
        }
}
f_e("Window");                                  //调用函数
echo "<p>";
f_e("Who_Window");                              //调用函数
```

```
echo "<p>";
f_e("temp_class");                         //调用函数
echo "<p>";
?>
</body>
</html>
```

把以上代码保存为“13-8.php”，然后在 PHP 执行环境中执行该文件，执行结果如图 13.8 所示。

通过以上实例及执行结果，用 class_exists()函数来判断类是否被定义是很方便的。而且该函数使用起来也很简单，此处不做过多解释。

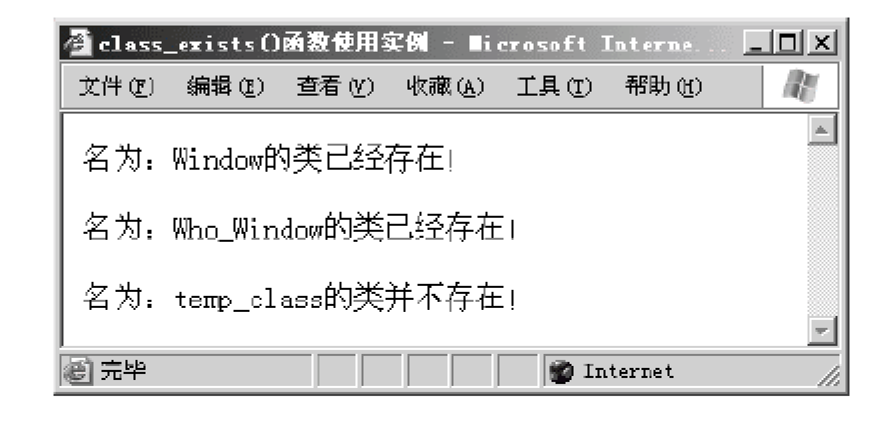


图 13.8　class_exists()函数使用实例执行结果

- ❑ array get_class_methods(mixed class_name)函数：返回由类的方法名组成的数组。调用该函数，将返回名称为 class_name 类的全部方法名。通过这一个函数就能了解到某个类的方法的情况。

了解一个类的方法名的详细情况也是很必要的。知道类的方法名，就可以对其引用。

下面通过一个实例来说明如何使用 get_class_methods()函数来获得类的全部方法名。代码如下：

```
<html>
<head>
<title>get_class_methods()函数使用实例</title>
</head>
<body>
<?
class Window                               //首先定义一个类
{
        var $state;                        //窗户的状态
        function close_window()            //关窗户方法
        {
                $this->state="close";      //窗户的状态为关
        }
        function open_window()             //开窗户方法
        {
                $this->state="open";       //窗户的状态为开
        }
}
$temp=get_class_methods("Window");
echo "类 Window 中的方法有以下几个：";
echo "<p>";
for($i=0;$i<count($temp);$i++)
{
        echo $temp[$i]."，";
}
?>
</body>
</html>
```

保存以上代码为“13-9.php”，然后在 PHP 执行环境下执行该文件，其执行结果如图 13.9 所示。

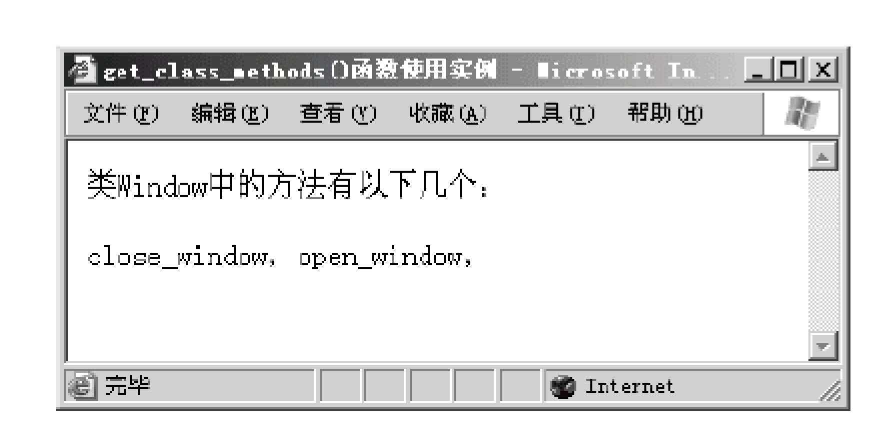


图 13.9　get_class_methods()函数使用实例执行结果

- ❑ array get_class_vars(string class_name)函数：调用该函数将会返回由类的默认公有属性组成的关联数组，此数组的元素以 varname=>value 的形式存在。

在使用一个类并为其定义对象之前也有必要了解类中变量的情况。下面通过实例来向读者介绍如何使用该函数来获取相应的内容。具体内容参看以下代码：

```
<html>
<head>
<title>get_class_vars()函数使用实例</title>
</head>
<body>
<?
class Dog                                    //首先定义一个类
{
      var $name;                             //狗的名字
      var $age;                              //狗的年龄
      var $birthday="3/15";                  //狗的生日
      var $sex="male";                       //狗的性别
}
$temp=array_keys(get_class_vars("Dog"));     //调用 get_class_vars()函数
echo "类 Dog 的属性有以下几个：";
echo "<p>";
print_r($temp);
?>
</body>
</html>
```

把以上代码保存为“13-10.php”，然后在 PHP 执行环境中运行该文件，执行结果如图 13.10 所示。

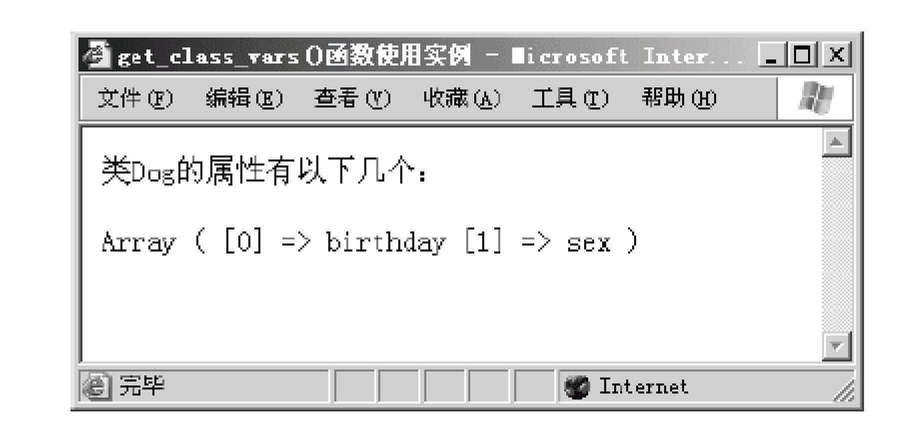


图 13.10　get_class_vars()函数使用实例执行结果

为什么只返回了两项内容呢？因为这里使用的 PHP 版本低于 4.2。如果在高于 PHP 4.2.0 的版本上执行，输出结果将会如下：

```
Array([0]=>name [1]=>age [2]=>birthday [3]=>sex)
```

- ❑ string get_class([object obj])函数：返回对象的类名。调用该函数将会返回对象实例 obj 所属类的名字。如果 obj 不是一个对象，则返回 False。

对象都是基于类创建的，一般情况下，知道了对象名就知道了相应的类名。但是如果对象特别多，也记不起是基于哪个类创建的情况下，可以使用该函数。

在使用此函数时有以下两种情况需要引起注意。

① 在 PHP 扩展库中定义的类返回其原始定义的名字。在 PHP 4 中，get_class()返回用户定义的类名的小写形式，但是在 PHP 5 中将返回类名定义时的名字，如同扩展库中的类名一样。

② 自 PHP 5 起，如果在对象的方法中调用，则 obj 为可选项。

下面通过一个实例来说明函数 get_class()的使用方法。代码如下：

```
<html>
<head>
<title>get_class ()函数使用实例</title>
</head>
<body>
<?
class Dog                              //首先定义一个类
{
    var $name;                         //狗的名字
    var $age;                          //狗的年龄
    var $birthday;                     //狗的生日
    var $sex;                          //狗的性别
}
$mydog=new Dog;                        //实例化一个对象
$mydog->name="wang";                   //定义名字
$mydog->age="two month";               //定义年龄
$mydog->birthday="3/15";               //定义生日
$mydog->sex="male";                    //定义性别
echo "对象 mydog 所属的类为：";
echo get_class($mydog);                //调用函数输出对象所属的类
?>
</body>
</html>
```

把以上代码保存为“13-11.php”，然后在 PHP 执行环境中执行该文件，其执行结果将会如图 13.11 所示。

通过上例及执行结果可以看出，调用 get_class()函数返回了对象$mydog 所属的类 Dog。从而说明，函数正确运行。

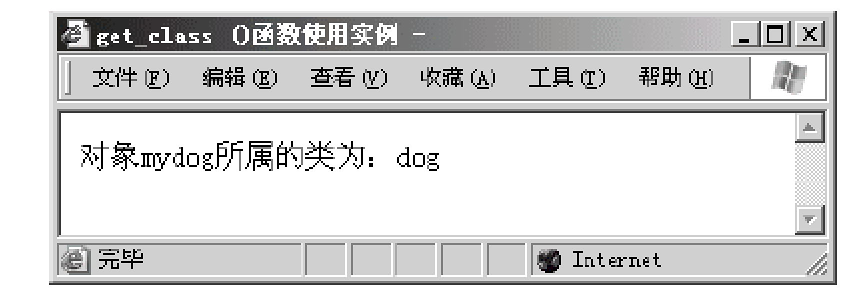


图 13.11　get_class()函数使用实例执行结果

- array get_declared_classes()函数：返回由已定义类的名字所组成的数组。调用该函数将会返回由当前脚本中已定义类的名字组成的数组。注意，此函数并没有任何参数。

由于 PHP 中有预定义类，即系统自动生成的类，所以使用该函数将会有几个预定义类存在于返回的数组中，分别为 stdClass、__PHP_Incomplete_Class、Directory。同时，由于有这些预定义类的存在，所以在创建类时不能使用这些预定义类名作为类的名称。

下面通过一个实例来说明函数 get_declared_classes()的使用方法。代码如下：

```
<html>
<head>
<title>get_declared_classes()函数使用实例</title>
</head>
```

```
<body>
<?
class Dog                                       //定义一个类 Dog
{
    var $name;                                  //狗的名字
    var $age;                                   //狗的年龄
    var $birthday;                              //狗的生日
    var $sex;                                   //狗的性别
}
class Window                                    //定义一个类 Window
{
    var $state;                                 //窗户的状态
    function close_window()                     //关窗户方法
    {
        $this->state="close";                   //窗户的状态为关
    }
    function open_window()                      //开窗户方法
    {
        $this->state="open";                    //窗户的状态为开
    }
}
class Who_Window extends Window                 //创建 Window 的子类 Who_Window
{
    var $owner;
    function close_window()                     //方法继承
    {
        $this->state="close";
        $this->owner="Jack";
    }
}
/*至此，此 PHP 文件中一共定义有 3 个类，分别是：Dog、Window、Who_Window。
现在调用 get_declared_classes()函数，看能否正确返回*/
$temp=get_declared_classes();                   //调用函数，把结果保存到变量中
for($i=0;$i<count($temp);$i++)                  //通过循环显示数组所有元素
{
    echo "数组的第".$i."个元素为：";
    echo $temp[$i];
    echo "<p>";
}
?>
</body>
</html>
```

保存以上代码为“13-12.php”，然后在 PHP 执行环境中运行该文件，其执行结果会如图 13.12 所示。

从图 13.12 可以看出，由于有预定义类的存在，所以函数首先返回了 6 个预定义类：stdClass、__PHP_Incomplete_Class、OverloadedTestClass、Directory、COM、VARIANT（由于 PHP 版本的不同，返回的预定义类可能也会有所不同），然后才返回 3 个用户自定义类：dog、window、who_window。

get_declared_classes()函数使用实例

数组的第0个元素为：stdClass
数组的第1个元素为：__PHP_Incomplete_Class
数组的第2个元素为：OverloadedTestClass
数组的第3个元素为：Directory
数组的第4个元素为：COM
数组的第5个元素为：VARIANT
数组的第6个元素为：dog
数组的第7个元素为：window
数组的第8个元素为：who_window

图 13.12　get_declared_classes()函数使用实例执行结果

array get_object_vars(object obj)函数：该函数返回由对象属性组成的关联数组。

在使用此函数有一点需要引起注意：在 PHP 4.2.0 之前的版本中，如果在 obj 对象实例中声明的变量没有被赋值，则它们将不会在返回的数组中。而在 PHP 4.2.0 以后的版本中，这些变量作为键名将被赋予 NULL 值。

下面给出一个实例来说明函数 get_object_vars()是如何获得对象属性的。代码如下：

```
<html>
<head>
<title>get_object_vars()函数使用实例</title>
</head>
<body>
<?
class Dog                                  //首先定义一个类
{
      var $name;                           //狗的名字
      var $age;                            //狗的年龄
      var $birthday;                       //狗的生日
      var $sex;                            //狗的性别
}
$mydog=new Dog;                            //实例化一个对象
$mydog->name="wang";                       //定义名字
$mydog->age="two month";                   //定义年龄
$mydog->birthday="3/15";                   //定义生日
$mydog->sex="male";                        //定义性别
$temp=get_object_vars($mydog);             //调用函数把对象的属性赋值给数组
print_r($temp);
?>
</body>
</html>
```

把以上代码保存为“13-13.php”，在 PHP 执行环境下运行该文件，其执行结果将会如图 13.13 所示。

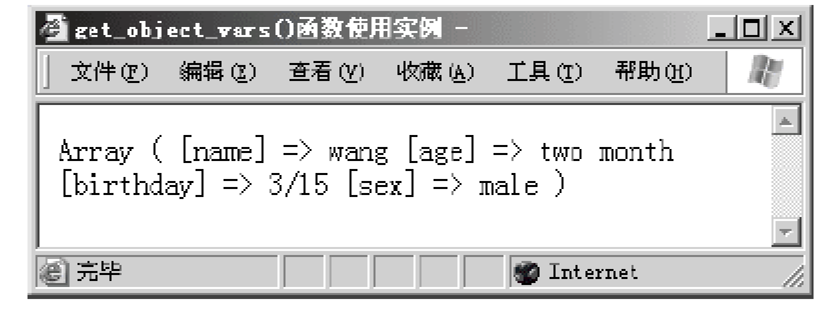


图 13.13　get_object_vars()函数使用实例执行结果

从如图 13.13 所示的执行结果中可以看出，函数返回了对象 Mydog 的所有属性及其值。这说明该函数 get_object_vars()正确运行。

- string get_parent_class([mixed obj])函数：返回对象或类的父类名。如果 obj 是对象，则返回对象实例 obj 所属类的父类名。如果 obj 是字符串，则返回以此字符串为名的类的父类名。此功能是在 PHP 4.0.5 中增加的。另外，自 PHP 5 起，如果在对象的方法内调用，则 obj 为可选项。

下面通过实例来说明如何使用 get_parent_class()函数来返回对象或类的父类名。代码如下：

```
<html>
<head>
<title>get_parent_class()函数使用实例</title>
</head>
<body>
<?
class Window                                //定义一个类 Window
{
    var $state;                             //窗户的状态
    function close_window()                 //关窗户方法
    {
        $this->state="close";               //窗户的状态为关
    }
    function open_window()                  //开窗户方法
    {
        $this->state="open";                //窗户的状态为开
    }
}
class Who_Window extends Window             //创建 Window 的子类 Who_Window
{
    var $owner;
    function close_window()                 //方法继承
    {
        $this->state="close";
        $this->owner="Jack";
    }
}
$my_who_window=new Who_Window;              //实例化一个对象
$temp1=get_parent_class("Who_Window");      //调用 get_parent_class()把结果赋值给变量
$temp2=get_parent_class($my_who_window);    //调用 get_parent_class()把结果赋值给变量
echo "类 Who_Window 的父类为：";
echo "<p>";
echo $temp1;                                //输出结果
echo "<p>";
echo "对象 my_who_window 所属类的父类为：";
echo "<p>";
echo $temp2;                                //输出结果
?>
</body>
</html>
```

把以上代码保存为“13-14.php”，在 PHP 执行环境中运行该文件，其执行结果如图 13.14 所示。

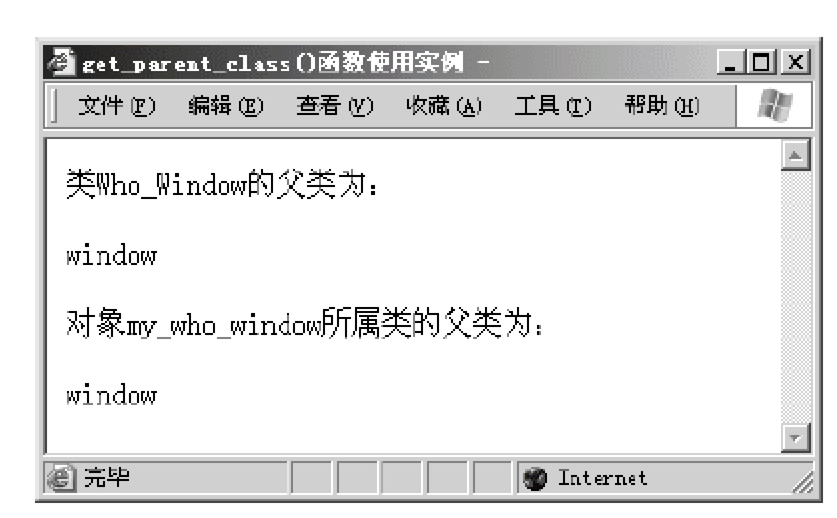


图 13.14　get_parent_class()函数使用实例执行结果

通过上例可以看出，第 1 次调用 get_parent_class()函数时使用的参数是字符串，即子类的名称，函数能正确返回父类的名称；第 2 次调用时使用的参数是对象名，函数返回了对象$my_who_window 所属的类 Who_Window 的父类 Window。通过两次调用可以看

出，函数 get_parent_class()不仅能返回子类的父类名称，也能返回对象所属子类的父类。

- bool is_subclass_of(object object,string class_name)函数：如果对象 object 所属类是类 class_name 的子类，则返回 True，否则返回 False。此函数的目的是判断某个对象是否属于某个类，或者是其父类。

下面通过一个实例来说明如何使用 is_sub_of()函数来进行对象与类的关系的判断。代码如下：

```
<html>
<head>
<title>is_subclass_of()函数使用实例</title>
</head>
<body>
<?
class Window                                        //首先定义一个类
{
        var $state;                                 //窗户的状态
        function close_window()                     //关窗户方法
        {
                $this->state="close";               //窗户的状态为关
        }
        function open_window()                      //开窗户方法
        {
                $this->state="open";                //窗户的状态为开
        }
}
class Who_Window extends Window                     //创建 Window 的子类 Who_Window
{
        var $owner;
        function close_window()                     //方法继承
        {
                $this->state="close";
                $this->owner="Jack";
        }
}
function is_sub_e($obj,$string)                     //创建一个基于 is_subclass_of()的自定义函数
{
        if(is_subclass_of($obj,$string))            //如果类名存在
        {
                echo "对象属于名为".$string."的类的子类的对象！ ";  //打印相应信息
        }
        else                                        //如果类不存在
        {
                echo "对象不属于名为".$string."的类的子类的对象！ ";//打印相应信息
        }
}
class Dog                                           //首先定义一个类
{
        var $name;                                  //狗的名字
        var $age;                                   //狗的年龄
```

```
        var $birthday;                                    //狗的生日
        var $sex;                                         //狗的性别
}
$my_window=new Who_Window;
is_sub_e($my_window,"Who_Window");                        //调用自定义函数
echo "<p>";
is_sub_e($my_window,"Window");                            //调用自定义函数
echo "<p>";
is_sub_e($my_window,"Dog");                               //调用自定义函数
?>
</body>
</html>
```

保存以上代码为“13-15.php”，然后在 PHP 执行环境中运行该文件，其执行结果将会如图 13.15 所示。

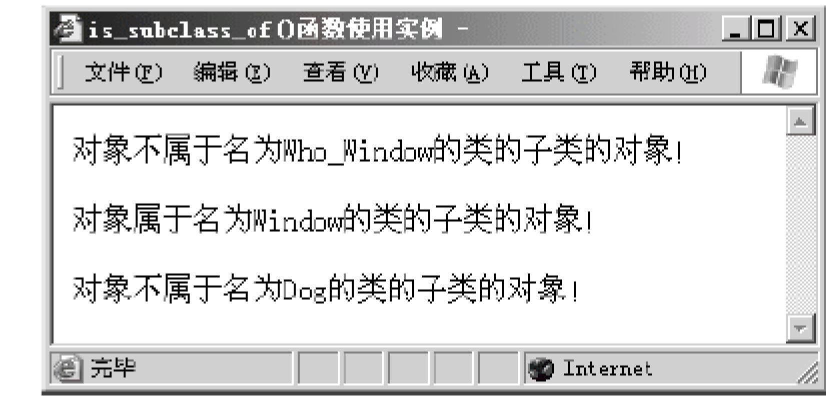


图 13.15　is_subclass_of()函数使用实例执行结果

从图 13.15 可以看出，由于对象$my_window 是基于类 Who_window 建立的，而类 Who_window 又是类 window 的子类，所以执行 is_subclass_of()函数，在列出的 3 个类中只有 window 类是符合条件的。所以图 13.15 中只有第 2 次输出了属于的信息，其他两次都输出不属于的信息。

- bool method_exists(object object,string method_name)函数：该函数检查 object 类的 method_name 方法是否存在。如果存在，函数返回 True， 反之则返回 False。

下面给出一个实例来说明函数 method_exists()是如何对对象的方法进行操作的。代码如下：

```
<html>
<head>
<title>method_exists()函数使用实例</title>
</head>
<body>
<?
class Window                                              //首先定义一个类
{
        var $state;                                       //窗户的状态
        function close_window()                           //关窗户方法
        {
                $this->state="close";                     //窗户的状态为关
        }
        function open_window()                            //开窗户方法
        {
                $this->state="open";                      //窗户的状态为开
        }
}
function method_e($obj,$string)                           //创建一个基于 method_exists 的自定义函数
{
        if(method_exists($obj,$string))                   //如果类名存在
```

```
        {
            echo "对象中名为".$string."的类已经存在！";          //打印相应信息
        }
        else                                                    //如果类不存在
        {
            echo "对象中名为".$string."的类并不存在！";          //打印相应信息
        }
    }
    $my_window=new Window;
    method_e($my_window,"open_window");                        //调用自定义函数
    echo "<p>";
    method_e($my_window,"close_window");                       //调用自定义函数
    echo "<p>";
    method_e($my_window,"temp_method");                        //调用自定义函数
    ?>
    </body>
    </html>
```

保存以上代码为“13-16.php”，在 PHP 执行环境下运行该文件，执行结果将会如图 13.16 所示。

从图 13.16 可以看出，通过调用基于 method_exists()函数的自定义函数，正确地对对象中的方法情况进行了判断。上例中，由于 open_window 方法与 close_window 方法均已经存在，所以返回存在的信息。而 temp_method 则没有存在，所以返回方法不存在的信息。

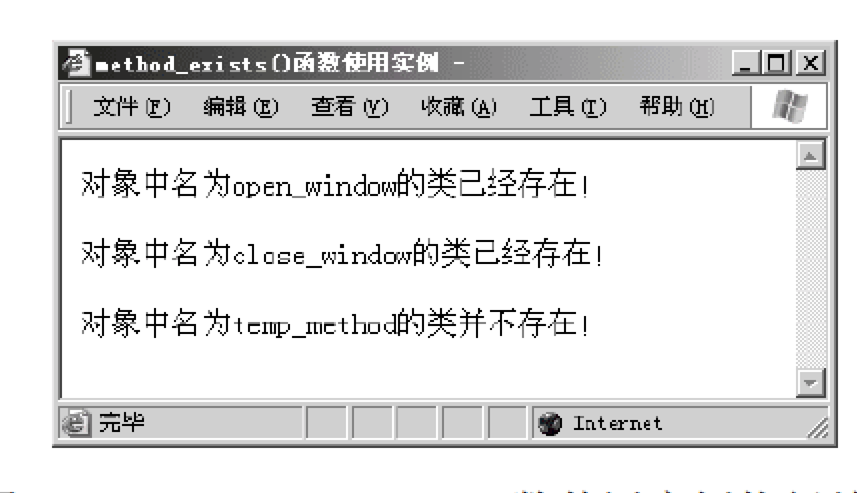


图 13.16　method_exists()函数使用实例执行结果

13.4　类的具体使用实例

本节将综合前面几节所学习的内容，做一个实例来实际地应用学的知识。本实例实现的功能是用 PHP 面向对象的特性写一个完全的购物车类。具体内容参看以下代码：

```
    <html>
    <head>
    <title>面向对象的具体使用——购物车类</title>
    </head>
    <body>
    <?
    class ShopCar                                              //类名购物车类
    {
        var $carName;                                          //属性购物车名字
        var $debug;                                            //属性
        function ShopCar($carName)                             //创建购物车方法（构造函数）
        {
            $this->carName=$carName;                           //将购物车命名为指定名称
            if(!isset($_SESSION[$carName]))                    //如果没有保存为 session
```

```
        {
            $_SESSION[$carName]=array();                    //把购物车名称保存到 session
        }
    }
    function addCar($type,$name,$val)                       //向购物车中增加商品方法
    {
        if(array_key_exists($type,$_SESSION[$this->carName]))     //如果类别已经存在
        {
            if(array_key_exists($name,$_SESSION[$this->carName][$type]))  //如果商品存在
            {
            if($this->debug)echo "<p>已有{$name}商品,不必增加<p>";       //输出相应信息
            return false;                                        //返回假
            }
            else                                                 //如果商品不存在
            {
                $_SESSION[$this->carName][$type][$name]=$val;   //在 session 中设定
            }
        }
        else                                                     //如果类别不存在
        {
            $_SESSION[$this->carName][$type]=array($name=>$val);  //在 sessuib 是设定
        }
        return true;                                             //返回真值
    }
    function editCar($type,$name,$var)                           //编辑购物车方法
    {
        if(!array_key_exists($name,$_SESSION[$this->carName][$type]))   //如果类别不存在
        {
            if($this->debug)echo "<p>没有{$name}商品,修改失败<p>";      //输出错误信息
            return false;                                        //返回假值
        }
        $_SESSION[$this->carName][$type][$name]=$var;            //在 session 中设定
        return true;
    }
    function delCarType($type)                                   //删除购物车类别
    {
        if(!array_key_exists($type,$_SESSION[$this->carName]))   //如果类别不存在
        {
            if($this->debug)echo "<p>没有{$type}类别,删除失败<p>";      //输出错误信息
            return false;                                        //返回假值
        }
        unset($_SESSION[$this->carName][$type]);                 //在 session 中删除
        return true;                                             //返回真值
    }
    function delCarPro($type,$name)                              //删除商品
    {
        if(!array_key_exists($name,$_SESSION[$this->carName][$type]))   //如果商品不存在
        {
            if($this->debug)echo "<p>没有{$name}商品,删除失败<p>";      //输出错误信息
            return false;                                        //返回假值
```

```
            }
            unset($_SESSION[$this->carName][$type][$name]);            //在 session 中删除
            return true;                                               //返回真值
        }
        function delCar()                                              //删除购物车
        {
            session_unregister($this->carName);                        //在 session 中注销
        }
        function getCarData()                                          //获取购物车数据
        {
            return $_SESSION[$this->carName];                          //通过 session 返回
        }
    }
    $my_car=new Shopcar("my_shopcar");                                 //创建对象时一定要带参数，
    否则会有出错提示。
    $my_car->addCar("水果","苹果",10);                                 //为购物车添加商品
    $my_car->addCar("水果","香蕉",5);
    $my_car->addCar("水果","梨",4);
    $my_car->addCar("蔬菜","白菜",10);
    $my_car->addCar("蔬菜","萝卜",3);
    $my_car->addCar("蔬菜","菠菜",12);
    $my_car->addCar("蔬菜","茄子",8);
    $temp=$my_car->getCarData();
    print_r ($temp);
    ?>
    </body>
    </html>
```

上例创建了一个功能完善的购物车类，既可以单独使用，也可以通过连接数据库，形成一个功能强大的在线购物平台。保存以上代码为“13-17.php”，然后在 PHP 执行环境中运行该文件，执行结果将会如图 13.17 所示。

通过图 13.17 可以看到，通过 addCar()方法向购物车中添加了部分商品，又通过 getCarData()方法获取了各类商品的相关信息。另外，该类中的删除商品、删除购物车等方法也都是非常有用的，有兴趣的读者可以继续完善上面的例子，以验证各种方法的正确性。这里就不再赘述。

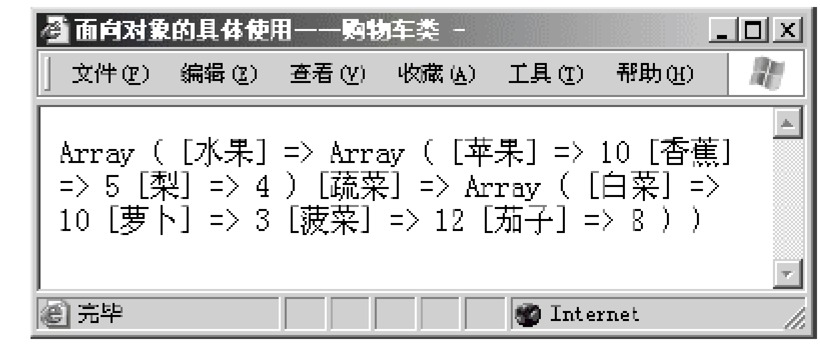


图 13.17　购物车类执行结果

13.5　本 章 小 结

本章主要带领读者学习了 PHP 的面向对象编程的特性。学习了什么是类，什么是对象，如何使用类，PHP 中与类及对象相关的函数。正如本章开始所提到的那样，编程语言的方法有面向对象与面向过程两种，具体采用那种方法，完全取决于用户的喜好。但是作者还是提议，在写小型程序，代码不多的情况下，采用面向过程的方法要好一点。同样，当开发大型项目，代码动辄成千上万行时，采用面向对象的编程方法能更好地提高工作效率。

第14章

使用 MySQL 数据库

- 关系型数据库基础
- 使用 PHP 对数据库进行操作
- 使用 PHP 对表进行操作
- 对表的高级查询
- 数据库使用实例

只有与数据库相结合，才能发挥动态网页编程语言的魅力，因为网上的众多应用都是基于数据库的。虽然 PHP 支持多种数据库，但 PHP 与 MySQL 可称得上是黄金组合。这一章就带领读者来学习 MySQL。本章内容包括关系型数据库基础、PHP 中与 MySQL 相关的常用函数、如何创建一个数据库、如何在 PHP 文档中执行 SQL 语句、MySQL 表的高级查询等内容。最后还将通过一个实例——班级学生管理系统，来巩固本章的知识。通过本章的学习会使读者能够熟练的写出基于数据库的 Web 应用程序。

14.1　关系型数据库基础

通常所说的数据库有两种类型：一种是以库、表为基础的关系型数据库；另一类是用文本文件、或者二进制文件作为数据载体的文本型数据库。这一节来介绍关系型数据库。

14.1.1　什么是关系型数据库

关系型数据库以行和列的形式存储数据，以便于用户理解。这一系列的行和列被称为表，一组表组成了数据库。用户用查询（Query）来检索数据库中的数据。一个 Query 是一个用于指定数据库中行和列的 SELECT 语句。

关系型表（TABLE）是构成关系型数据库（DATABASE）的基础。表是由这样的结构构成，用列（COLS）表示的字段与用行（ROWS）表示的记录。这样说还是有点抽象，举一个实际的例子来说明这个问题，如某一个班级全体学生的考试成绩表是一个表。用列来表示每一个科目，用行来表示每一个学生的成绩情况。这样，这个表里面的每一个科目，如语文、数学、英语、物理、化学、政治、历史等科目的成绩就可以被看作是表字段。而记录每一个学生分数的行，如学生张三、李四、王五、赵六这些信息就可以被看作是表的记录。

14.1.2　关系型数据库的功能

数据库最基本的功能就是作为数据存储的载体。所以，存储各种各样的数据是数据库的最基本的功能。当然除了这个最基本的功能之外，关系型数据库还具有如下功能。

- 复杂的数据计算。把数据存放于数据库后并不是已经完成任务了，加入数据只是数据库应用的开始。可以利用数据库的强大计算功能来实现复杂的数据计算。
- 数据统计。有的时候还需要对存放于数据库之中的数据进行某一类别的统计。如果这项工作靠人力来完成，工作量是相当可观的。而对于大型的数据库来说几乎是不可能完成的。而依靠数据库则可以在很短时间内完成对数据的统计工作。
- 数据检索。有时还需要对数据进行搜索。在关系型数据库中执行 select 语句能够在很短的时间内检索出所要的结果。
- 其他功能。除此之外，关系型数据库还有其他一些功能，如构建搜索引擎、游戏服务器存放游戏玩家数据等。

14.2　PHP 中与 MySQL 数据库相关的函数

这一节先来学习 PHP 中的与 MySQL 数据库操作相关的函数。通过对这些函数的学习，会使读者对在 PHP 中如何操作数据库有一个比较全面的了解。

下面就分别为读者逐个介绍 PHP 中与 MySQL 数据库操作相关的常用函数（如果读者觉得下面的内容枯燥无味可以先简单浏览一下，跳过这一节，直接看后面的内容，遇到相关的函数再回来查会比较好）。

- int mysql_affected_rows([resource link_identifier])函数。调用此函数将返回前一次 MySQL 操作所影响的记录行数。此操作包括 insert、update、delete 3 项查询查询的操作。
- bool mysql_close([resource link_identifier])函数。关闭 MySQL 连接。mysql_close()关闭指定的连接标识所关联的、到 MySQL 服务器的非持久连接。如果没有指定 link_identifier，则关闭上一个打开的连接。通常不需要使用 mysql_close()来关闭连接，因为已打开的非持久连接会在脚本执行完毕后自动关闭。
- resource mysql_connect([string server[,string username[,string password[,bool new_link[, int client_flags]]]]])函数。打开一个到 MySQL 服务器的连接。该函数的参数如下：server，可用的 MySQL 服务器。可以包括端口号，例如"hostname:port"，或者到本地套接字的路径，例如对于 localhost 的":/path/to/socket"。如果在 PHP.INI 中指令 mysql.default_host 未定义（默认情况），则默认值是'localhost:3306'。username，用户名，默认值是服务器进程所有者的用户名。password，密码，默认值是空密码。new_link，如果用同样的参数第 2 次调用 mysql_connect()，将不会建立新连接，而将返回已经打开的连接标识。参数 new_link 改变此行为并使 mysql_connect()总是打开新的连接，甚至当 mysql_connect()曾在前面被用同样的参数调用过。client_flags，该参数可以是以下常量的组合，MYSQL_CLIENT_SSL，MYSQL_CLIENT_COMPRESS，MYSQL_CLIENT_IGNORE_SPACE 或 MYSQL_CLIENT_INTERACTIVE。这几个参数的意义分别如表 14.1 所示。

表 14.1　MySQL 客户端常量

常　　量	说　　明
MYSQL_CLIENT_COMPRESS	使用压缩的通讯协议
MYSQL_CLIENT_IGNORE_SPACE	允许在函数名后留空格位
MYSQL_CLIENT_INTERACTIVE	允许设置断开连接之前所空闲等候的interactive_timeout时间（代替wait_timeout）
MYSQL_CLIENT_SSL	使用SSL加密。本标志仅在MySQL客户端库版本为4.x或更高版本时可用。在PHP 4和Windows版的PHP 5安装包中绑定的都是3.23.x

函数的返回值，如果成功则返回一个 MySQL 连接标识，失败则返回 False。

该函数经过了几次版本更新，详细情况如下说明。

PHP 4.3.0 版本，添加了 client_flags 参数；PHP 4.2.0 版添加了 new_link 参数。PHP 3.0.10 版对 server 添加":/path/to/socket"路径的支持。PHP 3.0.0 版对 server 添加":port"可选端口号的支持。

- bool mysql_create_db(string database name[,resource link_identifier])函数。调用该函数将尝试在指定的连接标识所关联的服务器上建立一个新数据库。参数 database_name 为想要创建的数据库名。参数 link_identifier 为 MySQL 的连接标识符。如果没有指定，默认使用最后被 mysql_connect()打开的连接。如果没有找到该连接，函数会尝试调用 mysql_connect()建立连接并使用它。如果发生意外，没有找到连接或无法建立连接，系统发出 E_WARNING 级别的警

告信息。返回值，如果成功则返回 True，失败则返回 False。

- bool mysql_data_seek(resource result,int row_number)函数。移动内部结果的指针,将指定的结果标识 result 所关联的 MySQL 结果内部的行指针移动到 row_number 所指定的行号。row_number 从 0 开始。row_number 的取值范围从 0 到（mysql_num_rows-1）。但是如果结果集为空（mysql_num_rows()==0），即数据库表中并没有任何记录，要将指针移动到 0 会失败并发出 E_WARNING 级的错误，同时 mysql_data_seek()将返回 False。本函数的参数为：result 为返回类型为 resource 的结果集。该结果集从 mysql_query()的调用中得到。row_number 为想要设定的新的结果集指针的行数。函数返回值，如果成功则返回 True，失败则返回 False。
- string mysql_db_name(resource result,int row[,mixed field]) 函数。调用该函数将取得 mysql_list_dbs()函数调用所返回的数据库名。参数 result 为 mysql_list_dbs()调用所返回的结果指针。row 为结果集中的行号。field 为某一个字段名。此函数的返回值，如果成功则返回数据库名，失败返回 False。如果返回了 False，用 mysql_error()来判断错误的种类。
- resource mysql_db_query(string database,string query[,resource link_identifier])函数。此函数发送一条 MySQL 查询。根据查询结果返回一个 MySQL 结果资源号，出错时返回 False。本函数会对 INSERT（插入）、UPDATE（更新）、DELETE（删除）3 项查询返回 True/False 来指示成功或失败。mysql_db_query()选择一个数据库并在其上执行查询。如果没有提供可选的连接标识，本函数会去找一个到 MySQL 服务器的已打开的连接，如果找不到已打开连接则会尝试无参数调用 mysql_connect()来建立一个。

注意： 此函数不会切换回先前连接到的数据库。换句话说，不能用此函数临时在另一个数据库上执行 sql 查询，只能手工切换回来。强烈建议用户在 sql 查询中使用 database.table 语法来替代此函数。自 PHP4.0.6 起不提倡使用此函数。不要用此函数，用 mysql_select_db()和 mysql_query()来替代。

- bool mysql_drop_db(string database_name[,resource link_identifier])函数。丢弃（删除）一个 MySQL 数据库。mysql_drop_db()尝试丢弃（删除）指定连接标识所关联的服务器上的整个数据库。如果成功则返回 True，失败则返回 False。为向下兼容也可以用 mysql_drop_db()函数，但不要这样做。

注意： 不提倡使用 mysql_drop_db()函数来删除数据库。最好用 mysql_query()提交一条 SQL DROP DATABASE 语句来替代。

警告： 如果 MySQL 扩展库是基于 MySQL4.x 客户端库建立的，则本函数不可用。

- int mysql_errno([resource link_identifier])函数。返回上一个 MySQL 操作中的错误信息的数字编码。如果没有出错则返回 0。从 MySQL 数据库后端来的错误不再发出警告，要用 mysql_errno()来提取错误代码。

注意： 本函数仅返回最近一次 MySQL 函数的执行（不包括 mysql_error()和 mysql_errno()）的错误代码，因此如果要使用此函数，确保在调用另一个 MySQL 函数之前检查它的值。如果指定了可选参数则用给定的连接提取错误代码。否则使用上一个打开的连接。

- string mysql_error([resource link_identifier])函数。返回上一个 MySQL 操作产生的文本错误信

息。如果没有出错则返回" "（空字符串）。如果没有指定连接资源号，则使用上一个成功打开的连接从 MySQL 服务器提取错误信息。从 MySQL 数据库后端来的错误不再发出警告，要用 mysql_error()来提取错误文本。

注意：和 mysql_errno()函数一样，本函数仅返回最近一次 MySQL 函数的执行（不包括 mysql_error()和 mysql_errno()的错误文本，因此如果要使用此函数，确保在调用另一个 MySQL 函数之前检查它的值。

- string mysql_escape_string(string unescaped_string)函数。本函数将 unescaped_string 转义，使之可以安全用于 mysql_query()。

注意：mysql_escape_string()函数并不转义“%”和“_”。

- array mysql_fetch_array(resource result[,int result_type])函数。返回根据从结果集取得的行生成的数组，如果没有更多行则返回 False。mysql_fetch_array()是 mysql_fetch_row()的扩展版本。除了将数据以数字索引方式储存在数组中之外，还可以将数据作为关联索引储存，用字段名作为键名。

注意：如果结果中的两个或两个以上的列具有相同字段名，最后一列将优先。要访问同名的其他列，必须用该列的数字索引或给该列起个别名。对有别名的列，不能再用原来的列名访问其内容。

- array mysql_fetch_assoc(resource result)函数。返回根据从结果集取得的行生成的关联数组，如果没有更多行则返回 False。用 mysql_fetch_assoc()和用 mysql_fetch_array()加上第 2 个可选参数 MySQL_ASSOC 的结果是完全相同的。它仅仅返回关联数组。这也是 mysql_fetch_array()函数初始的工作方式。如果在关联索引之外还需要数字索引，还是要用 mysql_fetch_array()。

注意：如果结果中的两个或两个以上的列具有相同字段名，最后一列将优先。要访问同名的其他列，要么用 mysql_fetch_row()来取得数字索引或给该列起个别名。参见 mysql_fetch_array()例子中有关别名说明。

有一点很重要必须指出，用 mysql_fetch_assoc()并不明显比用 mysql_fetch_row()慢，而且还提供了明显更多的值。

注意：本函数返回的字段名是区分大小写的。

- object mysql_fetch_field(resource result[,int field_offset])函数。从结果集中取得列信息并作为对象返回。如果没有指定字段偏移量，则下一个尚未被 mysql_fetch_field()取得的字段将会被提取。

对象的属性有以下几项分别为 name 属性为列名、table 属性为该列所在的表名、max_length 为该列最大长度、如果该列不能为 NULL 则 not_null 属性为 1、如果该列是 primary key 则 primary_key 属性 1、如果该列是 unique key 则 unique_key 属性为 1、如果该列是 non-unique key 则 multiple_key 为 1、如果该列是 numeric 则 numeric 为 1、如果该列是 BLOB 则 blob 属性为 1、type 属性为该列的类型、如果该列是无符号数则 unsigned 属性为 1、如果该列是 zero-filled 则 zerofill 属性为 1。

注意：本函数返回的字段名是区分大小写的。

- ❑ array mysql_fetch_lengths(resource result)函数。取得结果集 result 中每个输出的长度。如果出错返回 False。mysql_fetch_lengths()将上一次 mysql_fetch_row()，mysql_fetch_array()和 mysql_fetch_object()所返回的每个列的长度储存到一个数组中，偏移量从 0 开始。
- ❑ object mysql_fetch_object(resource result)函数。从结果集 result 中取得一行作为对象。如果没有更多行则返回 False。mysql_fetch_object()函数和 mysql_fetch_array()函数有一定的类似，只有一点区别就是返回的是一个对象而不是数组。间接的也意味着只能通过字段名来访问数组，而不是偏移量（数字是合法的属性名）。

注意：本函数返回的字段名是区分大小写的。

- ❑ array mysql_fetch_row(resource result)函数。返回根据所取得的行生成的数组，如果没有更多行则返回 False。mysql_fetch_row()从和指定的结果标识关联的结果集中取得一行数据并作为数组返回。每个结果的列储存在一个数组的单元中，偏移量从 0 开始。依次调用 mysql_fetch_row()将返回结果集中的下一行，如果没有更多行则返回 False。

注意：本函数返回的字段名是区分大小写的。

- ❑ string mysql_field_flags(resource result,int field_offset)函数。该函数返回指定字段的字段标志。每个标志都用一个单词表示，之间用一个空格分开，因此可以用 explode()把字符串分割到数组中。
- ❑ int mysql_field_len(resource result,int field_offset)函数。返回指定字段的长度。
- ❑ string mysql_field_name(resource result,int field_index)函数。该函数返回指定字段索引的字段名。其中 result 参数必须是一个合法的结果标识符，field_index 是该字段的数字偏移量。

注意：field_index 从 0 开始。如第 3 个字段的索引值其实是 2，第 4 个字段的索引值是 3，以此类推。本函数返回的字段名是区分大小写的。

- ❑ int mysql_field_seek(resource result,int field_offset)函数。用指定的字段偏移量检索。如果下一个 mysql_fetch_field()的调用不包括字段偏移量，则会返回本次 mysql_field_seek()中指定的偏移量的字段。
- ❑ string mysql_field_table(resource result,int field_offset)函数。取得指定字段所在的表名。
- ❑ string mysql_field_type(resource result,int field_offset)函数。取得结果集中指定字段的类型。该函数和 mysql_field_name()函数相似。参数完全相同，但该函数返回的是字段类型而不是字段名。字段类型有“int”、“real”、“string”、“blob”以及其他类型。
- ❑ bool mysql_free_result(resource result)函数。释放结果标识符 result 所占用的内存。该函数仅需要在考虑到返回很大的结果集时会占用多少内存时调用。在脚本结束后所有关联的内存都会被自动释放。如果成功释放内存则返回 True，失败则返回 False。
- ❑ string mysql_get_client_info()函数。调用该函数将取得 MySQL 客户端信息，函数返回一个字符串，该字符串代表了 MySQL 的版本。
- ❑ string mysql_get_host_info([resource link_identifier])函数。取得 MySQL 主机信息。mysql_get_host_info()返回一个字符串说明了连接 link_identifier 所使用的连接方式，包括服务

器的主机名。如果省略 link_identifier，则使用上一个打开的连接。

- int mysql_get_proto_info([resource link_identifier])函数。mysql_get_proto_info()返回 link_identifier 所使用的协议版本。如果省略 link_identifier，则使用上一个打开的连接。
- string mysql_get_server_info([resource link_identifier])函数。该函数返回 link_identifier 所使用的服务器版本。如果省略 link_identifier，则使用上一个打开的连接。
- string mysql_info([resource link_identifier])函数。mysql_info()返回通过给定的 link_identifier 所进行的最新一条查询的详细信息。如果没有指定 link_identifier，则假定为上一个打开的连接。

mysql_info()对以下列出的所有语句返回一个字符串。对于其他任何语句返回 False。字符串的格式取决于给出的语句。起作用的语句如下所示：

INSERT INTO SELECT 插入语句；

INSERT INTO VALUES 插入语句；

LOAD DATA INFILE 载入数据语句；

ALTER TABLE 改变表结构语句；

UPDATE 更新记录语句。

- int mysql_insert_id([resource link_identifier])函数。调用该函数将返回给定的 link_identifier 中上一步 INSERT 查询中产生的 AUTO_INCREMENT 的 ID 号。如果没有指定 link_identifier，则使用上一个打开的连接。如果上一查询没有产生 AUTO_INCREMENT 的值，则 mysql_insert_id()将会返回 0。如果需要保存该值以后使用，要确保在产生了值的查询后立即调用 mysql_insert_id()函数。

注意：MySQL 中的 SQL 函数 LAST_INSERT_ID()总是保存着最新产生的 AUTO_INCREMENT 值，并且不会在查询语句之间被重置。

警告：mysql_insert_id()将 MySQL 内部的 C API 函数 mysql_insert_id()的返回值转换成 long（PHP 中命名为 int）。如果 AUTO_INCREMENT 的列的类型是 BIGINT，则 mysql_insert_id()返回的值将可能不正确。可以在 SQL 查询中用 MySQL 内部的 SQL 函数 LAST_INSERT_ID()来替代以取得上一步 INSERT 操作产生的 ID。

- resource mysql_list_dbs([resource link_identifier])函数。调用该函数将返回一个结果指针，包含了当前 MySQL 进程中所有可用的数据库。用 mysql_tablename()函数来遍历此结果指针，或者任何使用结果表的函数。
- resource mysql_list_fields(string database_name,string table_name[,resource link_identifier])函数。调用该函数取得给定 table_name 表名的信息。参数是数据库名和表名。返回一个结果指针，可以用于 mysql_field_flags()，mysql_field_len()，mysql_field_name()和 mysql_field_type()。

注意：此函数已过时。最好用 mysql_query()来发出一条 SHOW COLUMNS FROM table [LIKE 'name'] 的 SQL 语句来代替。

- resource mysql_list_processes([resource link_identifier])函数。mysql_list_processes()返回一个结果指针，说明了当前服务器的线程。
- resource mysql_list_tables()函数。该函数接受一个数据库名并返回和 mysql_query()函数很相似

的一个结果指针。用 mysql_tablename()函数来遍历此结果指针，或者任何使用结果表的函数。database 参数是需要被取得其中的表名的数据库名。如果失败函数返回 False。

注意：该函数已经被删除了，请不要再使用该函数。可以用命令 mysql_query 发送一条 SQL 指令 SHOW TABLES FROM DATABASE 来实现该函数的功能。

- int mysql_num_fields(resource result)函数。调用该函数将会返回结果集中字段的数目。
- int mysql_num_rows(resource result)函数。调用此函数将会返回结果集中行的数目。此命令仅对 SELECT 语句有效。要取得被 INSERT 插入、UPDATE 更新或者 DELETE 删除查询所影响到的行的数目，用 mysql_affected_rows()函数。
- resource mysql_pconnect(string server[,string username[,string password[,int client_flags]]]))函数。如果成功则返回一个正确的 MySQL 持久连接标识符，出错则返回 False。mysql_pconnect()函数建立一个到 MySQL 服务器的连接。如果没有提供可选参数，则使用如下默认值：server='localhost:3306'，username=服务器进程所有者的用户名，password=空密码。client_flags 参数可以是以下常量的组合：MySQL_CLIENT_COMPRESS，MySQL_CLIENT_IGNORE_SPACE 或者 MySQL_CLIENT_INTERACTIVE。server 参数也可以包括端口号，例如"hostname:port"，或者是本机套接字的路径，例如":/path/to/socket"。

注意：对":port"端口的支持是 PHP 的 3.0B4 版添加的。对本机套接字的路径":/path/to/socket"的支持是 PHP 3.0.10 版添加的。可选参数 client_flags 自 PHP4.3.0 版起作用。此种连接称为“持久的”。

mysql_pconnect()和 mysql_connect()非常相似，但有两个主要区别。

首先，当连接的时候本函数将先尝试寻找一个在同一个主机上用同样的用户名和密码已经打开的（持久）连接，如果找到，则返回此连接标识而不打开新连接。

其次，当脚本执行完毕后到 SQL 服务器的连接不会被关闭，此连接将保持打开以备以后使用（mysql_close()不会关闭由 mysql_pconnect() 建立的连接）。

注意：此种连接仅能用于模块版本的 PHP。

- bool mysql_ping([resource link_identifier])函数。mysql_ping()函数检查到服务器的连接是否正常。如果断开，则自动尝试连接。本函数可用于空闲很久的脚本来检查服务器是否关闭了连接，如果有必要则重新连接上。如果到服务器的连接可用则 mysql_ping()返回 True，否则返回 False。
- resource mysql_query(string query[,resource link_identifier])函数。mysql_query()向与指定的连接标识符关联的服务器中的当前活动数据库发送一条查询。如果没有指定 link_identifier，则使用上一个打开的连接。如果没有打开的连接，本函数会尝试无参数调用 mysql_connect()函数来建立一个连接并使用。查询结果会被缓存。

注意：查询字符串不应以分号结束。

另外，mysql_query()仅对 SELECT，SHOW，EXPLAIN 或 DESCRIBE 语句返回一个资源标识符，如果查询执行不正确则返回 False。对于其他类型的 SQL 语句，mysql_query()在执行成功时返回 True，出错时返回 False。非 False 的返回值意味着查询是合法的并能够被服务器执行。这并不说明任何有关

影响到的或返回的行数。很有可能一条查询执行成功了但并未影响到或并未返回任何行。

如果没有权限访问查询语句中引用的表时，mysql_query()也会返回 False。

假定查询成功，可以调用 mysql_num_rows()来查看对应于 SELECT 语句返回了多少行，或者调用 mysql_affected_rows()来查看对应于 DELETE，INSERT，REPLACE 或 UPDATE 语句影响到了多少行。

仅对 SELECT，SHOW，DESCRIBE 或 EXPLAIN 语句 mysql_query()才会返回一个新的结果标识符，可以将其传递给 mysql_fetch_array()和其他处理结果表的函数。处理完结果集后可以通过调用 mysql_free_result()来释放与之关联的资源，尽管脚本执行完毕后会自动释放内存。

- string mysql_real_escape_string(string unescaped_string[,resource link_identifier])函数。本函数将 unescaped_string 中的特殊字符转义，并计及连接的当前字符集，因此可以安全用于 mysql_query()。

注意：mysql_real_escape_string()并不转义“%”和“_”。

- mixed mysql_result(resource result,int row[,mixed field])函数。调用该函数将返回 MySQL 结果集中一个单元的内容。字段参数可以是字段的偏移量或者字段名，或者是字段表点字段名（tablename.fieldname）。如果给列起了别名（'select foo as bar from...'），则用别名替代列名。

当作用于很大的结果集时，应该考虑使用能够取得整行的函数（在下边指出）。这些函数在一次函数调用中返回了多个单元的内容，比 mysql_result()快得多。此外注意在字段参数中指定数字偏移量比指定字段名或者 tablename.fieldname 要快得多。

注意：调用 mysql_result()不能和其他处理结果集的函数混合调用。

- bool mysql_select_db(string database_name[,resource link_identifier])函数。该函数将选择指定的 MySQL 数据库。如果成功则返回 True，失败则返回 False。mysql_select_db()设定与指定的连接标识符所关联的服务器上的当前激活数据库。如果没有指定连接标识符，则使用上一个打开的连接。如果没有打开的连接，本函数将无参数调用 mysql_connect()来尝试打开一个并使用。
- string mysql_stat([resource link_identifier])函数。mysql_stat()返回当前服务器状态。

注意：mysql_stat()目前只返回 uptime，threads，queries，open tables，flush tables 和 queries per second。要得到其他状态变量的完整列表，只能使用 SQL 命令 SHOW STATUS。取得当前系统状态。

- string mysql_tablename(resource result,int i)函数。mysql_tablename()接受 mysql_list_tables()返回的结果指针以及一个整数索引作为参数并返回表名。可以用 mysql_num_rows()函数来判断结果指针中的表的数目。用 mysql_tablename()函数来遍历此结果指针，或者任何处理结果表的函数。
- int mysql_thread_id([resource link_identifier])函数。mysql_thread_id()返回当前线程的 ID。如果连接丢失了并用 mysql_ping()重新连接上，线程 ID 会改变。这意味着不能取得线程的 ID 后保存起来备用。当需要的时候再去获取之。
- resource mysql_unbuffered_query(string query[,resource link_identifier])函数。该函数向 MySQL 发送一条 SQL 查询 query，但不像 mysql_query()那样自动获取并缓存结果集。一方面，这在

处理很大的结果集时会节省可观的内存。另一方面，可以在获取第一行后立即对结果集进行操作，而不用等到整个 SQL 语句都执行完毕。当使用多个数据库连接时，必须指定可选参数 link_identifier。

PHP 中的 MySQL 函数就为读者介绍到这里。实际上这些函数中，有一些是不太常用的。读者不必完全掌握，如果在实际操作中见到这样的函数，知道它们是如何起作用的就行了。但对于一些常用的函数则必须熟练掌握，因为熟练掌握其中的常用函数是学好数据库的前提。

14.3　数据库操作

这一节，就来学习实际对数据库的操作。内容包括如何连接数据服务器，如何显示目前的数据库，如何创建一个数据库，如何连接已经存在的库，如何在库中创建表及如何在表中插入记录等。

14.3.1　连接 MySQL 服务器

所有对 MySQL 数据库的操作都要在主机提供 MySQL 服务的前提下进行。连接提供了此项服务的主机是进行数据库操作的前提。所以在进行数据库操作前必须要连接到 MySQL 服务器。

在 PHP 编程环境下要连接到 MySQL 服务器，需要使用函数 mysql_connect()。该函数有 3 个参数，第 1 个参数为主机名；第 2 个参数为用户名；第 3 个参数为该用户的密码。此函数连接到 MySQL 服务器，如果成功返回 True，反之返回 False。

下面通过一个实例来说明，如何使用 mysql_connect()函数来连接到 MySQL 服务器。

```
<html>
<head>
<title>连接 MySQL 服务器</title>
</head>
<body>
<?php
$db_host=localhost;                    //MySQL 服务器名
$db_user=root;                         //MySQL 用户名
$db_pass="";                           //MySQL 用户对应密码
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
echo "成功连接到服务器";               //如果连接成功显示信息
?>
</body>
</html>
```

保存以上代码为“14-1.php”。然后开启 APACHE 服务及 MySQL 服务（本章实例均需要两种服务共同支持，下同），在 PHP 执行环境下运行该文件，其执行结果将会如图 14.1 所示。

上面代码中 mysql_connect()的作用是连接到 MySQL 服务器，它带有 3 个 string 参数。第 1 个参数指主机名，第 2 个参数是用户名，第 3 个参数是用户密码。这些信息都由 MySQL 服务提供者提供。

后面的 or die()指如果不能正确连接 MySQL 服务器的返回信息。

第 2 句很明显是输出了一句连接成功的信息。

执行以上操作，如果正确无误就能正确连接到 MySQL 服务器了。其执行结果如图 14.1 所示。如果关闭 MySQL 服务就会导致无法连接到服务器则程序会输出像这样的信息：



图 14.1　连接 MySQL 服务器执行结果

不能连接到服务器 Can't connect to MySQL server on 'localhost' (10061)。

关于如何连接到服务器就先介绍到这里。

14.3.2　连接到服务器并显示可用数据库

连接到 MySQL 服务器上后，下一步就是了解当前数据库的状况。因为只有了解服务器上的可用数据库信息，才能使用户决定下一步要如何操作。下面讲解如何显示可用的数据库。要显示服务器上的数据库情况使用 mysql_list_dbs()函数。该函数有一个参数，此参数为已经建立的服务器连接。

下面通过一个实例来说明如何使用 mysql_list_dbs()函数来显示服务器上的可用数据库。具体内容请参看以下代码：

```
<html>
<head>
<title>显示数据库</title>
</head>
<body>
<?php
$db_host=localhost;                        //MySQL 服务器名
$db_user=root;                             //MySQL 用户名
$db_pass="";                               //MySQL 用户对应密码
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
$db_list=mysql_list_dbs($link);            //显示数据库，参数为上一步的服务器连接
while($db=mysql_fetch_object($db_list))    //通过循环遍历返回的结果集
{
    echo $db->Database;                    //显示数据库名，注意大小写
    echo "<p>";
}
?>
</body>
</html>
```

保存以上代码为“14-2.php”。然后在 PHP 执行环境下执行该文件，执行结果将会如图 14.2 所示。

由于此时服务器上只有两个数据库，MySQL 库与 TEST 库，所以显示两条记录。



图 14.2　显示数据库执行结果

14.3.3　在服务器上创建新的数据库

对数据库进行操作时，系统提供的数据库通常是有限的，所以需要在服务器上建立新的数据库。下面介绍如何在服务器上创建一个新的数据库。要创建新的数据库，要使用 mysql_create_db()函数。mysql_create_db()函数的使用格式如下所示：

```
mysql_creaet_db(string database name[,resource link_identifier])
```

该函数包含两个参数，第 1 个参数为要创建的数据库名，第 2 个参数为可选参数，为已经连接的服务器的连接标识。

下面通过一个实例来介绍如何使用该函数在服务器上创建新的数据库。

```
<html>
<head>
<title>创建新的数据库</title>
</head>
<body>
<?php
$db_host=localhost;                     //MySQL 服务器名
$db_user=root;                          //MySQL 用户名
$db_pass="";                            //MySQL 用户对应密码
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
if(mysql_create_db("database1",$link))  //使用 mysql_create_db()创建数据库
echo "数据库创建成功";                  //如果创建成功显示信息
echo "<p>";
echo "当前服务器上的所有数据库为：";
echo "<p>";
$db_list=mysql_list_dbs($link);         //显示数据库
while($db=mysql_fetch_object($db_list)) //通过循环遍历返回的结果集
{
    echo $db->Database;                 //显示数据库名，注意大小写
    echo "<p>";
}
?>
</body>
</html>
```

保存以上代码为“14-3.php”，然后在 PHP 执行环境下运行该文件，其执行结果如图 14.3 所示。

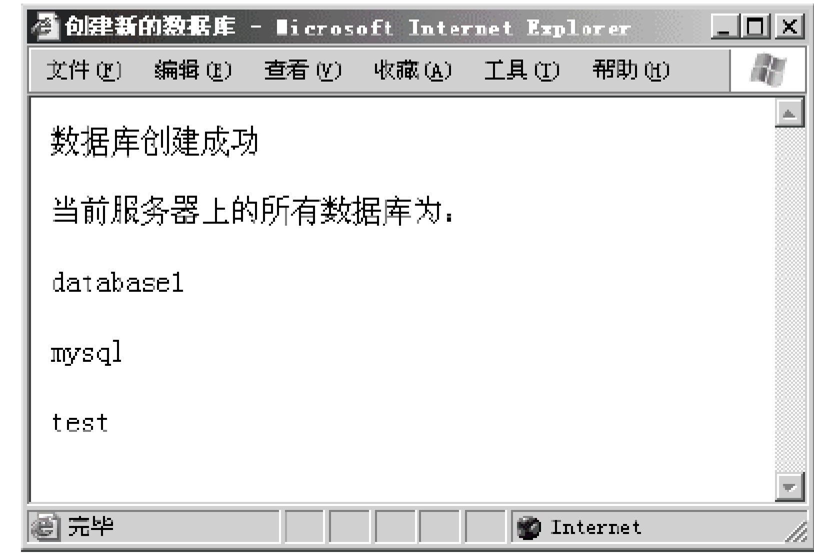


图 14.3　创建新的数据库执行结果

由于该函数已经废弃，所以建议在创建数据库时使用 mysql_query()来发送一条 sql 语句达到创建数据库的目的。具体作法请参看如下代码：

```
<html>
<head>
```

```
<title>使用 SQL 语句创建新的数据库</title>
</head>
<body>
<?php
$db_host=localhost;                         //MySQL 服务器名
$db_user=root;                              //MySQL 用户名
$db_pass="";                                //MySQL 用户对应密码
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
$sql="CREATE DATABASE data2";               //创建数据库的 SQL 语句
if(mysql_query($sql,$link))                 //发送 SQL 语句
echo "数据库创建成功";                       //如果创建成功显示信息
echo "<p>";
echo "当前服务器上的所有数据库为：";
echo "<p>";
$db_list=mysql_list_dbs($link);             //显示数据库
while($db=mysql_fetch_object($db_list))     //通过循环遍历返回的结果集
{
    echo $db->Database;                     //显示数据库名，注意大小写
    echo "<p>";
}
?>
</body>
</html>
```

把以上代码保存为“14-4.php”，然后在 PHP 执行环境下运行该文件，其执行结果如图 14.4 所示。

从以上两个例子及执行结果可以看出，使用 SQL 语句同样能达到创建数据库的目的。要注意的是创建数据库的 SQL 语句的语法：为“CREATE DATABASE”后面加上想要创建的数据库名称即可。然后使用 mysql_query()来发送 SQL 语句。来达到创建数据库的目的。

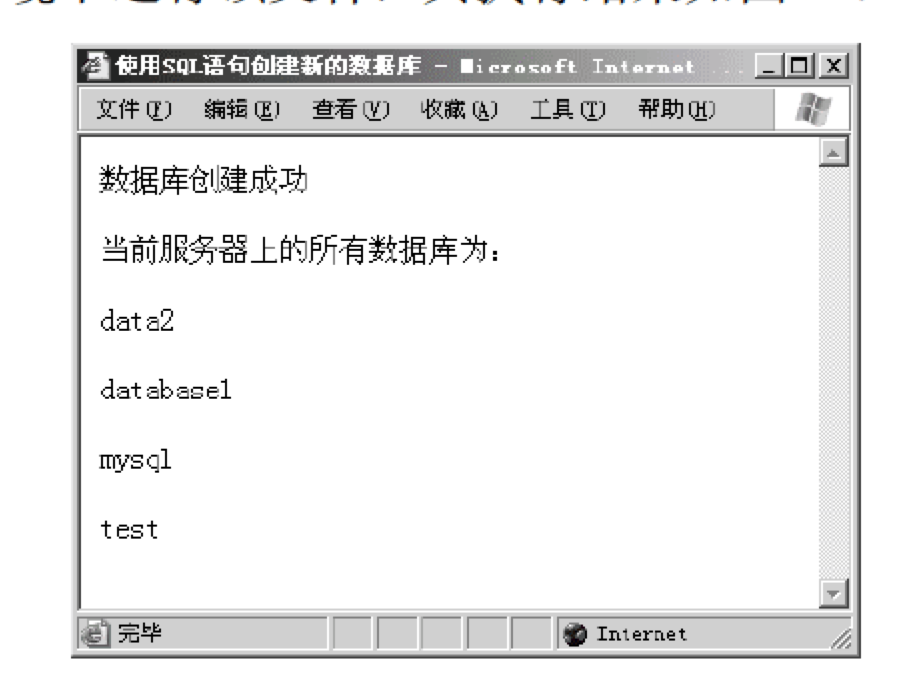


图 14.4　使用 SQL 语句创建新的数据库执行结果

14.3.4　在选定数据库里创建表

通过前面的介绍读者能了解到，表是数据库的主要构成元素。有了数据库还必须有特定的表才能发挥作用，因为基本上所有的数据库操作都是在表中进行的。这一小节就来介绍如何在数据库里创建表。

注意：服务器上 MySQL 数据库是运行 MySQL 服务的关键库，不要尝试对该库进行操作，更不要对里面的表进行添加、删除、修改等操作。

要想在数据库中创建表，要使用 mysql_query()函数发送 SQL 语句来实现。

创建表的 SQL 语句的语法如下所示：

```
create table table_name (column_name data 无效 {identity |null|not null}, …)
```

其中参数 table_name 为表名和 column_name 为列名即字段名必须满足用户数据库中的识别器（identifier）的要求，参数 data 无效是一个标准的 SQL 类型或由用户数据库提供的类型。用户要使用 non-null 从句为各字段输入数据。

create table 还有一些其他选项，如创建临时表和使用 select 子句从其他的表中读取某些字段组成新表等。还有，在创建表是可用 PRIMARY KEY、KEY、INDEX 等标识符设定某些字段为主键或索引等。

书写上要注意：

- 在一对圆括号里的列出完整的字段清单。
- 字段名间用逗号隔开。
- 字段名间的逗号后要加一个空格。
- 最后一个字段名后不用逗号。
- 所有的 SQL 陈述都以分号“;”结束。

如下面的这一句：

```
$sql="create table table1(id int(5) not null auto_increment primary key, name varchar(12) not null,mail varchar(30) not null,phone varchar(14) not null,address varchar(30) not null)";
```

下面的 SQL 语句定义了一个表，表有这样几个字段：id、name、mail、phone、address 等。并且还为每个字段指定了类型，并且其中的 id 字段有 primart 属性和 auto_increment 这样的属性。其中 primary 指明该字段为表的主键（有唯一值）；而 auto_increment 则指明这个字段是自动增加的。

下面，就使用该 SQL 语句在 test 库中创建一个名为 test1 的表。具体内容参看以下代码：

```
<html>
<head>
<title>在库中创建新表</title>
</head>
<body>
<?php
$db_host=localhost;                         //MySQL 服务器名
$db_user=root;                              //MySQL 用户名
$db_pass="";                                //MySQL 用户对应密码
$db_name="test";                            //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);            //选择相应的数据库，这里选择 test 库
//下面的$sql 就为创建表的 SQL 语句
$sql="create table test1(id int(5) not null auto_increment primary key, name varchar(12) not null,mail varchar(30) not null,phone varchar(14) not null,address varchar(30) not null)";
if(mysql_query($sql,$link))                 //发送 SQL 语句执行创建表的操作
echo "表 test1 创建成功";                    //如果创建成功显示信息
echo "<p>";
echo "当前".$db_name."上的所有数据表为：";
echo "<p>";
$table_list=mysql_list_tables($db_name,$link); //显示数据库中的表
while($row=mysql_fetch_row($table_list))    //通过循环遍历返回的结果集
{
    echo $row[0];                           //显示表名
```

```
        echo "<p>";
    }
    ?>
    </body>
    </html>
```

把以上代码保存为“14-5.php”。在 PHP 执行环境中运行该文件，其执行结果如图 14.5 所示。

上例中使用了 mysql_list_tables()函数。该函数的作用是显示库中所有的表。就像前面讲到的创建数据库函数 mysql_create_db()一样，该函数也已经被废弃，所以不推荐使用，可以用 mysql_query()发送一条 SQL 语句来达到和函数一样的目的。显示所有数据表的 SQL 语句是：“show tables”。

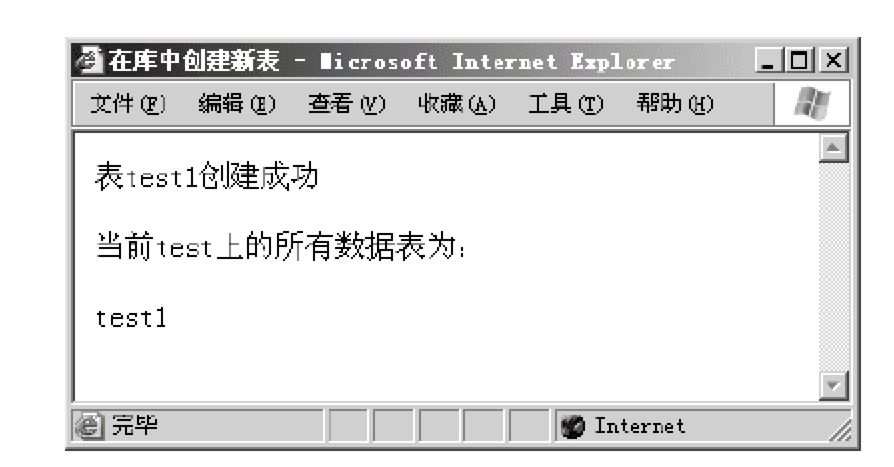


图 14.5　在库中创建新表执行结果

下面把上面的代码做一下调整，把其中的

```
$table_list=mysql_list_tables($db_name,$link);
```

一句更换为

```
$table_list=mysql_query("show tables",$link);
```

然后，重新执行修改过的 PHP 文件，可以看到执行结果是完全相同的（再次执行时要把创建表相关几行都注释掉，或者再换个表名进行创建才行。因为对已经存在的表进行创建就会导致错误）。

对如何创建表，这一节就讲到这里。这一节不仅学习了如何创建表，还学习了如何显示库中所有的表。14.3.5 节就讲解如何对表内数据进行操作了。

14.3.5　如何删除已经存在的库和表

如果确认数据库或者里面的表已经不再需要，就可以将库或者表删除。如何删除库或者表呢？

删除一个库有两种方法，一种是通过 PHP 中的 MySQL 函数 mysql_drop_db()；另一种是通过 mysql_query()函数发送 SQL 语句 drop database database_name。

而删除一个表只有一种方法就是通过 mysql_query()函数发送 SQL 语句 drop table table_name。

下面分别来介绍如何删除表及库。

先来看如何删除一个表。删除表是靠 mysql_query()函数来发送 SQL 语句 drop table_name 来实现，其中的 table_name 指代将要被删除的表名。下面的例子演示了如何删除 test 库中的表 test1。具体内容参看以下代码：

```
<html>
<head>
<title>在库中删除表</title>
</head>
<body>
<?php
$db_host=localhost;                        //MySQL 服务器名
$db_user=root;                             //MySQL 用户名
```

```
$db_pass="";                                //MySQL 用户对应密码
$db_name="test";                            //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);            //选择相应的数据库，这里选择 test 库
echo "当前".$db_name."上的所有数据表为：";  //先显示表删除之后再显示以作比较
echo "<p>";
$table_list=mysql_list_tables($db_name,$link); //显示数据库中的表
while($row=mysql_fetch_row($table_list))    //通过循环遍历返回的结果集
{
    echo $row[0];                           //显示表名
    echo "<p>";
}
$sql="drop table test1";                    //删除表的 sql 语句
if(mysql_query($sql,$link))                 //执行 SQL 语句
echo "表 test1 已经被删除！";
echo "<p>";
echo "当前".$db_name."上的所有数据表为：";  //再次显示删除之后的表以作比较
echo "<p>";
$table_list=mysql_list_tables($db_name,$link); //显示数据库中的表
while($row=mysql_fetch_row($table_list))    //通过循环遍历返回的结果集
{
    echo $row[0];                           //显示表名
    echo "<p>";
}
?>
</body>
</html>
```

把以上代码保存为“14-6.php”，然后在 PHP 执行环境下运行该文件（在执行前要确认已经建立了 test 库及 test1 表），其执行结果将会如图 14.6 所示。



图 14.6　在库中删除表执行结果

通过图 14.6 可以看到，在执行删除表的 SQL 语句之前，对库进行遍历，库中有表 test1。在执行完删除表的 SQL 语句后，再次遍历库已经不存在 test1 表了。说明该表已经被成功删除。

接下来，学习如何删除库。删除库可以使用 mysql_drop_db()函数，或者用 mysql_query()函数发送 SQL 请求 drop database database_name，都能够实现。这里使用后一种方法来实现。下面的代码实现了把服务器上的名为 data2 的数据库进行删除（在执行前请确保 MySQL 服务器中有 data2 这个数据库）。具体内容参看以下代码：

```
<html>
<head>
<title>删除服务器上的数据库</title>
</head>
```

```
<body>
<?php
$db_host=localhost;                          //MySQL 服务器名
$db_user=root;                               //MySQL 用户名
$db_pass="";                                 //MySQL 用户对应密码
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
$db_list=mysql_list_dbs($link);              //先显示数据库以与删除后作比较
while($db=mysql_fetch_object($db_list))      //通过循环遍历返回的结果集
{
    echo $db->Database;                      //显示数据库名，注意大小写
    echo "<p>";
}
$sql="drop database data2";                  //删除数据库的 sql 语句
if(mysql_query($sql,$link))                  //执行 SQL 语句
echo "数据库 data2 已经被删除！";
echo "<p>";
echo "当前服务器上的所有数据库为：";         //再次显示删除后的数据库以作比较
echo "<p>";
$db_list=mysql_list_dbs($link);              //显示数据库
while($db=mysql_fetch_object($db_list))      //通过循环遍历返回的结果集
{
    echo $db->Database;                      //显示数据库名，注意大小写
    echo "<p>";
}
?>
</body>
</html>
```

保存以上代码为“14-7.php”。在 PHP 执行环境中执行该文件，其执行结果将会如图 14.7 所示。



图 14.7　删除服务器上的数据库执行结果

通过对照 SQL 执行前后对服务器上数据库的遍历可以看到，执行完 SQL 语句后，名为 data2 的数据库被删除了。说明以上程序正常运行。

如何删除数据库及表，这一节就介绍到这里。

在使用对数据库操作时有一点需要注意，由于当前的服务提供商一般对于一个用户只提供一个数据库，所以普通用户只能在该库内对表进行操作，无权创建或者删除数据库。所以在执行相关操作时是不会有结果的。如果在本机上进行调试，就没有这个限制。

14.4 节将为读者介绍，如何对表进行操作。

14.4　对 MySQL 表进行操作

在创建了库，并为库创建了相应的表后，就可以对表中的数据进行操作了。其实对表的操作无非就是插入、浏览、修改、删除这几项。而这几项操作都可以通过 mysql_query()发送相关的 SQL 语句来实现。这一节就来为读者介绍如何实现对数据表中记录的插入、浏览、修改及删除操作。认真学过本节内容后，读者能对如何操作数据表，有一个充分的认识。

14.4.1　执行 INSERT INTO 语句插入记录

一个表在创建之后是没有任何记录的，只有在其中插入记录才能发挥表的作用。要对表中插入记录，可以使用 mysql_query()函数发送 SQL 请求。SQL 语句的内容如下所示：

```
insert into table_name(field01,field02…) values(values1,value2…)
```

其中 table_name 是表的名称，field01、field02 是指表的字段名称，values1、values2 是指想要插入的记录中对应字段的值。

假如有名为 test1 的表，它的结构如下所示：

```
id int(5) not null auto_increment primary key,
name varchar(12) not null,
mail varchar(30) not null,
phone varchar(14) not null,
address varchar(30) not null
```

可以看出，该表有 5 列（字段）。分别是 ID（编号）、NAME（名称）、MAIL（电子信箱）、PHONE（电话）、ADDRESS（地址）。如果想要把名字为“张三”的记录插入 test1 表中。就应该执行这样的操作：

```
insert into test1(name,mail,phone,address) values("张三","zhangsan@homail.com","1234567","某省某市某区某街")
```

为什么没有给 ID 赋值？因为字段 ID 为关键字段，并且具有自动增加 1 的属性，所以不用为其单独赋值。第 1 条记录的 ID 号就自动为 1，以后每插入一条，ID 号就会自动向后增加。

下面，就把以上的分析归结到实际的例子中来实际执行一下，以检测能否正确插入记录（在执行前，请确保 test 库中有 test1 这个表，如果没有，可以看上一节的相关内容，在库里创建一个）。

```
<html>
<head>
<title>在表中插入记录</title>
</head>
<body>
<?php
$db_host=localhost;                          //MySQL 服务器名
$db_user=root;                               //MySQL 用户名
```

```
$db_pass="";                               //MySQL 用户对应密码
$db_name="test";                           //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);           //选择相应的数据库，这里选择 test 库
$sql="select * from test1";                //先执行 SQL 语句显示所有记录以与插入后相比较
$result=mysql_query($sql,$link);           //使用 mysql_query()发送 SQL 请求
echo "当前表中的记录有：";
echo "<p>";
while($row=mysql_fetch_array($result))     //遍历 SQL 语句执行结果把值赋给数组
{
    echo $row[id]."，";                    //显示 ID
    echo $row[name]."，";                  //显示姓名
    echo $row[mail]."，";                  //显示邮箱
    echo $row[phone]."，";                 //显示电话
    $row[address]."，";                    //显示地址
    echo "<p>";
}
//以下 SQL 语句为插入记录操作
$sql="insert into test1(name,mail,phone,address) values('张三','zhangsan@homail.com','1234567','某省某市某
区某街')";
if(mysql_query($sql))                      //判断并执行 SQL 语句
echo "记录已经成功插入";
echo "<p>";
$sql="select * from test1";                //先执行 SQL 语句显示所有记录以与插入后相比较
$result=mysql_query($sql,$link);           //使用 mysql_query()发送 SQL 请求
echo "当前表中的记录有：";
echo "<p>";
while($row=mysql_fetch_array($result))     //再次遍历结果
{
    echo $row[id]."，";                    //显示 ID
    echo $row[name]."，";                  //显示姓名
    echo $row[mail]."，";                  //显示邮箱
    echo $row[phone]."，";                 //显示电话
    echo $row[address]."，";               //显示地址
    echo "<p>";
}
?>
</body>
</html>
```

把以上代码保存为“14-8.php”。在 PHP 执行环境下运行该文件，执行结果将会如图 14.8 所示。

通过图 14.8 可以看到，在第一次对表进行浏览操作时，由于没有任何记录，所以并没有任何值返回。当执行完 SQL 语句后再对表进行遍历时就有了一条记录，并且显示出了记录的内容。说明在表中顺利插入了一条记录。



图 14.8　在表中插入记录执行结果

下面对上例中相关的一些函数作用做简单的介绍。

mysql_fetch_array()：返回根据从结果集取得的行生成的数组。

上例中执行了 select 查询，对表的内容进行浏览，结果保存到$result。而使用 mysql_fetch_array()将返回根据从$result 取得的行生成一个数组。上面显示记录内容用到的$row[id]、$row[anme]等数组元素，就是由 mysql_fetch_array()函数获得的。

14.4.2　执行 SELECT 查询

显示表中的内容，或者对表中特定内容进行搜索时都需要对表进行查询。所以对表的查询操作也是用到最多的 SQL 命令之一。对表中记录进行浏览要使用 mysql_query()函数发送 SQL 请求。查询是使用最多的 SQL 命令。查询数据库需要凭借结构、索引和字段类型等因素。select 查询的基本语法如下所示：

```
SELECT [STRAIGHT_JOIN] [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [HIGH_PRIORITY]
[DISTINCT | DISTINCTROW | ALL]
select_expression，...
[INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
[FROM table_references
][WHERE where_definition]
[GROUP BY col_name，...]
[HAVING where_definition]
[ORDER BY {unsigned_integer | col_name | formula} ][ASC | DESC] ，...]
[LIMIT ][offset，] rows]
[PROCEDURE procedure_name] ]
```

太多的语法，暂时还用不到，以后在高级查询中专门为读者讲解。现在只是要浏览表中所有的记录。只需要使用以下指令：

```
select * from tablename
```

“*”号表示所有内容，所以这一句表示查询 tablename 表中的所有记录。这在前一小节对数据表插入记录时已经见到过。

下面通过一个实例来说明如何使用 select 查询获得表中的记录。

```
<html>
<head>
<title>浏览表中记录</title>
</head>
<body>
<center>
<?php
$db_host=localhost;                             //MySQL 服务器名
$db_user=root;                                  //MySQL 用户名
$db_pass="";                                    //MySQL 用户对应密码
$db_name="test";                                //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
```

```
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);              //选择相应的数据库，这里选择 test 库
$sql="select * from test1";                   //先执行 SQL 语句显示所有记录以与插入后相比较
$result=mysql_query($sql,$link);              //使用 mysql_query()发送 SQL 请求
echo "当前表中的记录有：";
echo "<table border=1>";                      //使用表格格式化数据
echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
while($row=mysql_fetch_array($result))        //遍历 SQL 语句执行结果把值赋给数组
{
    echo "<tr>";
    echo "<td>".$row[id]."</td>";             //显示 ID
    echo "<td>".$row[name]." </td>";          //显示姓名
    echo "<td>".$row[mail]." </td>";          //显示邮箱
    echo "<td>".$row[phone]." </td>";         //显示电话
    echo "<td>".$row[address]." </td>";       //显示地址
    echo "</tr>";
}
echo "</table>";
?>
</center>
</body>
</html>
```

把以上代码保存为“14-9.php”。在 PHP 执行环境中执行该文件，其执行结果将会如图 14.9 所示。



图 14.9　浏览表中记录执行结果

14.4.3　使用表单扩展添加记录功能

虽然 insert into 是执行添加记录操作的核心，但不能插入一条记录都把 14-8.php 中的内容修改一遍执行一遍。所以应该采用互动的形式，用户输入什么数据，就插入相应的记录。这一小节，把 14.4.1，14.4.2 节的内容结合起来，再利用 Web 表单，来实现用户输入记录的添加，来做一个简易的通讯录。其中的表还使用 test 库中的 test1 表。

显示记录的页面，只需要把 14-9.php 作简单修改，把标题换一下并加入指向添加记录前台即下面的 14-11.php 的连接，另存为 14-10.php，以备后用。

下面制作添加记录的前台，主要是使用了 Web 表单，并把 form 的 action 属性指向目标页面。具体内容参看以下代码：

```
<html>
<head>
<title>简易通讯录添加记录前台</title>
</head>
<body>
<script language="javascript">
function Juge(theForm)
{
    if (theForm.name.value == "")
```

```
        {
                alert("请输入姓名！");
                theForm.name.focus();
                return (false);
        }
        if (theForm.phone.value == "")
        {
                alert("请输入电话号码！");
                theForm.phone.focus();
                return (false);
        }
        if (theForm.address.value == "")
        {
                alert("请输入地址！");
                theForm.address.focus();
                return (false);
        }
}
</script>
<center>
<h1>简易通讯录添加记录前台</h1>
<p>
<a href="14-10.php">返回首页</a>
<table border=1>
<form action="14-12.php" method="post" onsubmit="return Juge(this)">
<tr>
<td>输入姓名：</td>
<td><input name="name" type="text"></td>
</tr>
<tr>
<td>输入邮箱：</td>
<td><input name="mail" type="text"></td>
</tr>
<tr>
<td>输入电话：</td>
<td><input name="phone" type="text"></td>
</tr>
<tr>
<td>输入地址：</td>
<td><input name="address" type="text"></td>
</tr>
<tr>
<td colspan="2"><center><input type=submit value="确认提交">
<input type=reset value="重新选择"></center></td>
</tr>
</form>
</table>
</center>
</body>
</html>
```

把以上代码保存为“14-11.php”。备用。

最后制作对输入记录的后台处理页面。添加记录的核心还是使用 INSERT INTO 这样一条 SQL 语句，把记录插入表中。下面的例子，还说明了一个问题，如何获取表单输入的数据，并把记录存入注册表，这是实现与用户互动的关键技术。

```
<?
/*下面内容为获取表单输入，并去掉 HTML 格式。
由于表单采用 post 方式传递数据，所以用 post 来获取输入*/
$name=htmlspecialchars($_POST['name']);            //获取姓名
$mail=htmlspecialchars($_POST['mail']);            //获取邮箱
$phone=htmlspecialchars($_POST['phone']);          //获取电话
$address=htmlspecialchars($_POST['address']);      //获取地址
$db_host=localhost;                                //MySQL 服务器名
$db_user=root;                                     //MySQL 用户名
$db_pass="";                                       //MySQL 用户对应密码
$db_name="test";                                   //要操作的数据库
$table_name="test1";                               //表名
$myconn=mysql_connect("$db_host","$db_user","$db_pass"); //连接服务器
mysql_select_db("$db_name",$myconn);               //选择操作库
$strSql="insert into $table_name(name,mail,phone,address) values ('$name','$mail','$phone','$address')";
    //对表进行插入操作
mysql_query($strSql,$myconn) or die("插入时出错".mysql_error());//发送 SQL 请求
?>
<html>
<head>
<title>简易通讯录添加记录处理</title>
</head>
<meta http-equiv="refresh" content="2; url=14-10.php">
<body>
已经成功添加记录，两秒后返回。
</body>
</html>
```

把以上代码保存为“14-12.php”。至此，整个简易通讯录程序就算是制作完毕了。到底能否实现，手动输入记录信息，并把记录插入到表中呢？下面就来运行一下整个程序，看执行结果。

首先运行 14-10.php，其执行结果将会如图 14.10 所示。



图 14.10　简易通讯录显示数据页执行结果

从图 14.10 的执行结果可见，显示出了原有的数据（这数据是在前面讲的内容中添加的）。接下来，

单击添加记录连接，相当于执行 14-11.php。单击连接后的结果如图 14.11 所示。

由于表中记录只有 5 项，而 ID 一项还是主键并且具有自动增加的属性。所以不必为其添加值。只需要输入图 14.12 所示的 4 项内容即可。按照提示输入相应的信息。这里输入以下简单信息：李四、lisi@mircosoft.com、2345678、北京市海淀区（读者也可以根据自己需要输入合适的信息）。输入完后单击“确认提交”按钮，将转到 14-12.php。

而 14-12.php 只对数据进行了处理，马上又自动跳转到了 14-10.php，如图 14.12 所示。

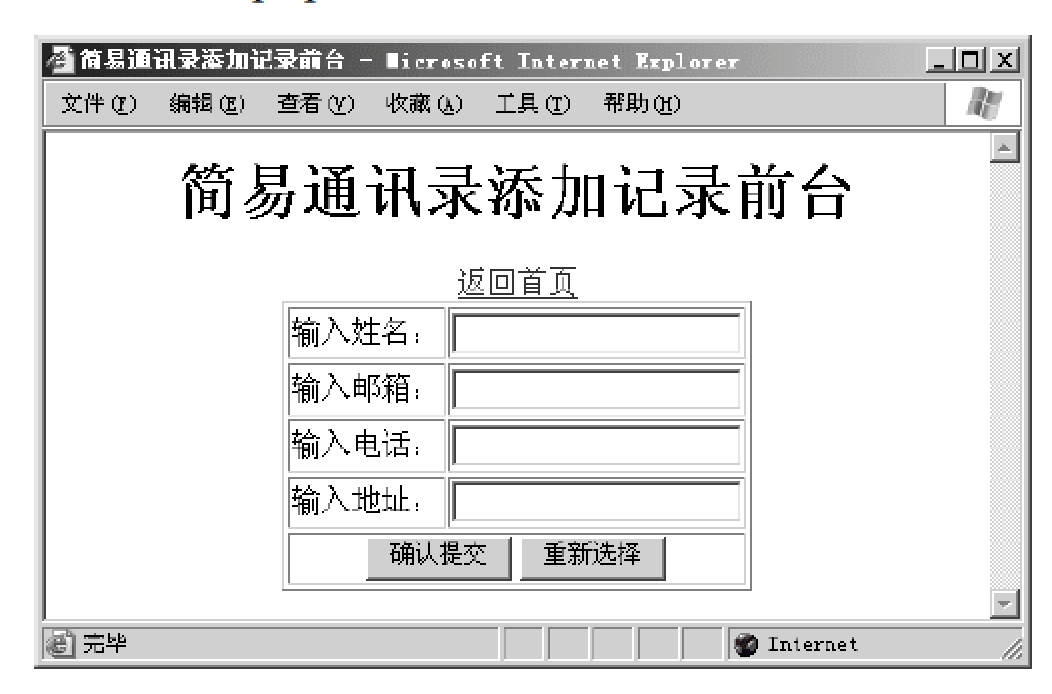


图 14.11　简易通讯录添加记录前台执行结果



图 14.12　插入后的数据显示页执行结果

通过图 14.12 可见，显示的记录中在原有的张三后，又显示了刚刚插入的李四的记录，说明记录已经被插到数据库之中。该简易通讯录程序测试通过。

14.4.4　执行 UPDATE 语句更新记录

记录在入库后，并不是一成不变的，有时可能需要对某些内容作相应的调整，这时就要用到 update 语句对记录进行更新（修改）了。update 语句的使用格式如下：

```
update table_name set field01="value1",field02="value02" where where_definition
```

其中的 table_name 为欲操作的表名。field01、field02 为字段名，而后面的 value1、value2 为想要更改为的内容。后面的 where_definition 为更改源记录的条件。比如想把 test1 表中张三的记录改为王五的，可以执行以下操作。

```
update test1 set name="王五",mail="wangwu@tom.com",phone="3456789",address="上海市高新开发区"
where name="张三";
```

后面的 where name="张三"如果改成 id=1 会更合适。因为字段 ID 是主键，并且具有唯一性，所以不可能出现重复。而 name 则不一样，如果有重名的“张三”则执行相应的操作会把所有的都进行更新。下面就把这段代码加入到实际的 PHP 文件中，看如何使用 UPDATE 语句来更改已经存在的记录。具体内容参看以下代码：

```
<html>
<head>
<title>更改表中记录</title>
</head>
<body>
```

```
<center>
<?php
$db_host=localhost;                          //MySQL 服务器名
$db_user=root;                               //MySQL 用户名
$db_pass="";                                 //MySQL 用户对应密码
$db_name="test";                             //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);             //选择相应的数据库，这里选择 test 库
function show_con()                          //把显示记录功能做成函数以便多次调用
{
    $link=mysql_connect(localhost,root,"");
    mysql_select_db(test);
    $sql="select * from test1";              //先执行 SQL 语句显示所有记录以与更改后相比较
    $result=mysql_query($sql,$link);         //使用 mysql_query()发送 SQL 请求
    echo "<table border=1>";                 //使用表格格式化数据
    echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
    while($row=mysql_fetch_array($result))   //遍历 SQL 语句执行结果把值赋给数组
    {
        echo "<tr>";
        echo "<td>".$row[id]."</td>";            //显示 ID
        echo "<td>".$row[name]." </td>";         //显示姓名
        echo "<td>".$row[mail]." </td>";         //显示邮箱
        echo "<td>".$row[phone]." </td>";        //显示电话
        echo "<td>".$row[address]." </td>";      //显示地址
        echo "</tr>";
    }
    echo "</table>";
}
echo "更新前的记录为：";
show_con();                                  //调用函数，显示表中所有记录
$sql="update test1 set name='王五',mail='wangwu@tom.com',phone='3456789',address='上海市高新开发区'
where id=1";                                 //创建更新记录 SQL 语句
mysql_query($sql,$link);                     //发送 SQL 请求
echo "更新后的记录为：";
show_con();                                  //再次调用函数，显示表中记录以做比较
?>
</center>
</body>
</html>
```

把以上代码保存为“14-13.php”，在 PHP 执行环境下运行该文件，执行结果如图 14.13 所示。



图 14.13　更改表中记录执行结果

从图 14.13 执行结果可以看到在执行更新数据操作之前，调用显示记录函数返回的第 1 条记录为张三。而执行完更新操作后，再次调用显示记录函数，返回的第 1 条记录已经成了

王五了。说明顺利地把表中记录号为 1 的记录进行了更改。

14.4.5　使用表单扩展更改记录功能

就像插入记录操作一样，不可能每更改一条记录就修改一次 14-13.php 中的内容，并执行一次该 PHP 文件。如果能与 Web 表单结合起来，实现用户输入什么内容，就更改什么内容，那样就会方便很多。下面，就来介绍，如何使用 Web 表单与更改数据记录的操作结合起来。这一节，要把 14.4.2 节所制作的显示记录文件作简单的修改。具体代码如下：

```
<html>
<head>
<title>浏览表中记录</title>
</head>
<body>
<center>
<?php
$db_host=localhost;                          //MySQL 服务器名
$db_user=root;                               //MySQL 用户名
$db_pass="";                                 //MySQL 用户对应密码
$db_name="test";                             //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);             //选择相应的数据库，这里选择 test 库
$sql="select * from test1";                  //先执行 SQL 语句显示所有记录以与插入后相比较
$result=mysql_query($sql,$link);             //使用 mysql_query()发送 SQL 请求
echo "当前表中的记录有：";
echo "<table border=1>";                     //使用表格格式化数据
echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td><td> </td></tr>";
while($row=mysql_fetch_array($result))       //遍历 SQL 语句执行结果把值赋给数组
{
      echo "<tr>";
      echo "<td>".$row[id]."</td>";          //显示 ID
      echo "<td>".$row[name]." </td>";       //显示姓名
      echo "<td>".$row[mail]." </td>";       //显示邮箱
      echo "<td>".$row[phone]." </td>";      //显示电话
      echo "<td>".$row[address]." </td>";    //显示地址
      echo "<td><a href='14-15.php?id=".$row[id]."'>修改</a></td>";
      echo "</tr>";
}
echo "</table>";
?>
</center>
</body>
</html>
```

可以看出，与 14-10.php 不同的是，为显示的表格增加了一列，列中显示指向修改记录前台页面的连接。把以上代码保存为“14-14.php”，以备后用。

下面制作修改记录的前台页面。该页面与 14-11.php 即插入记录的前台基本相似，不同的是由于是修改已经存在的记录，所以要能显示原有的记录内容。具体修改内容参看以下代码：

```
<html>
<head>
<title>简易通讯录修改记录前台</title>
</head>
<body>
<script language="javascript">
function Juge(theForm)
{
	if (theForm.name.value == "")
	{
		alert("请输入姓名！");
		theForm.name.focus();
		return (false);
	}
	if (theForm.phone.value == "")
	{
		alert("请输入电话号码！");
		theForm.phone.focus();
		return (false);
	}
	if (theForm.address.value == "")
	{
		alert("请输入地址！");
		theForm.address.focus();
		return (false);
	}
}
</script>
<center>
<h1>简易通讯录修改记录前台</h1>
<p>
<a href="14-14.php">返回首页</a>
<table border=1>
<form action="14-16.php" method="post" onsubmit="return Juge(this)">
<?
$link=mysql_connect(localhost,root,"");
mysql_select_db(test,$link);                //选择相应的数据库，这里选择 test 库
$sql="select * from test1 where id=$_GET[id]"; //只显示请求 ID 号的内容
$result=mysql_query($sql);
$row=mysql_fetch_array($result);            //把结果赋值给数组
?>
<input type="hidden" name="id" value="<?echo $row[id]?>">
<tr>
<td>输入姓名：</td>
<td><input name="name" type="text" value="<?echo $row[name]?>"></td>
</tr>
```

```
<tr>
<td>输入邮箱：</td>
<td><input name="mail" type="text" value="<?echo $row[mail]?>"></td>
</tr>
<tr>
<td>输入电话：</td>
<td><input name="phone" type="text" value="<?echo $row[phone]?>"></td>
</tr>
<tr>
<td>输入地址：</td>
<td><input name="address" type="text" value="<?echo $row[address]?>"></td>
</tr>
<tr>
<td colspan="2"><center><input type=submit value="确认提交">
<input type=reset value="重新选择"></center></td>
</tr>
</form>
</table>
</center>
</body>
</html>
```

可以看出，上面代码与 14-11.php 的不同，就是多了一项，从表中查找出想要修改的记录的原始内容。并把相应的值赋值给相应的表单元素。同时，该表单中多了一个隐藏的表单元素 ID，它指向记录的 ID，作为提交表单并且修改记录的条件。

把以上代码另存为“14-15.php”，以备后用。

接下来制作修改记录的后台操作页面。具体代码如下所示：

```
<?
/*下面内容为获取表单输入，并去掉 HTML 格式。
由于表单采用 post 方式传递数据，所以用 post 来获取输入*/
$name=htmlspecialchars($_POST['name']);           //获取姓名
$mail=htmlspecialchars($_POST['mail']);           //获取邮箱
$phone=htmlspecialchars($_POST['phone']);         //获取电话
$address=htmlspecialchars($_POST['address']);     //获取地址
$id=$_POST['id'];                                 //获取 ID
$db_host=localhost;                               //MySQL 服务器名
$db_user=root;                                    //MySQL 用户名
$db_pass="";                                      //MySQL 用户对应密码
$db_name="test";                                  //要操作的数据库
$table_name="test1";                              //表名
$myconn=mysql_connect("$db_host","$db_user","$db_pass"); //连接服务器
mysql_select_db("$db_name",$myconn);              //选择操作库
$strSql="update $table_name set name='$name',mail='$mail',phone='$phone',address='$address' where
id=$id";                                          //对表进行修改操作
mysql_query($strSql,$myconn) or die("插入时出错".mysql_error());//发送 SQL 请求
?>
<html>
<head>
```

```
<html>
<head>
<title>简易通讯录修改记录处理</title>
</head>
<meta http-equiv="refresh" content="2; url=14-14.php">
<body>
已经成功修改记录，两秒后返回。
</body>
</html>
```

可以看出以上代码与 14-12.php 也基本类似，除了$SQL 语句内容有所不同，其他地方都相同。把以上代码保存为“14-16.php”，以备后用。

这 3 个文件创建完毕，开始测试更改功能是否起作用。第 1 步先在 PHP 执行环境中运行 14-14.php。其结果如图 14.14 所示。

从图 14.14 可以看出与图 14.10 的不同之处就在于表格多出一列修改的连接。单击该连接，进入相应的修改页面。这里单击第 1 条记录后面的连接，就连接到 14-15.php，其执行结果如图 14.15 所示。



图 14.14　浏览表中记录带修改连接执行结果

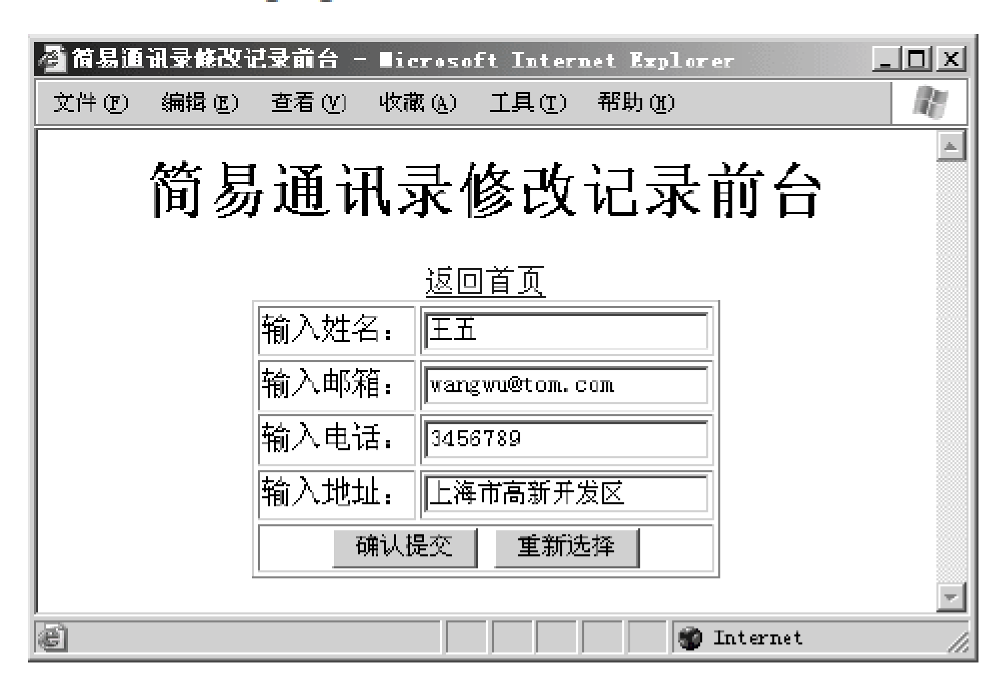


图 14.15　简易通讯录修改记录前台执行结果

从图 14.15 可以看出，连接到此页，显示出了原来存在的记录的内容。下面对该记录进行适当修改。为了便于对照，只把其中的电话由 3456789 改为 1234567。把上海市高新开发区，修改为上海市南京路（读者也可以根据自己需要进行相应的修改）。其他内容保持不变。输入相应内容后，单击“确认提交”按钮。将转到 14-16.php 对数据进行处理。执行结果如图 14.16 所示。



图 14.16　修改记录后台处理执行结果

由于 14-16.php 使用了页面自动跳转，所以在执行完修改记录操作后，直接跳转到了 14-14.php。不过，比较图 14.14 与图 14.16 可以发现，ID 号为 1 的记录中的电话号码和地址都已经发生了改变。由原来的内容，变成了在图 14.15 中输入的内容。说明对数据的重新记录操作顺利执行。

这一节用 Web 表单扩展修改表中记录的功能就介绍到这里。如果把这一小节所介绍的内容与 14.4.3

小节所介绍的功能结合起来，就是一个功能完善（添加记录、修改记录）的简易通讯录程序了。

14.4.6 执行 DELETE 语句删除记录

数据表中的内容如果不再需要，为了节省空间，把多余的不再需要的记录从表中删除是有必要的。这一节为读者介绍，如何删除表中的记录。

要想删除表中的不再需要的记录，可以使用 mysql_query()函数发送一条 delete 的 SQL 语句。delete 的使用语法如下所示：

```
delete from table_name where t field01="value1"
```

其中的 table_name 为欲删除的记录所在的表名。后面的 where 子句表示执行操作的条件。field01 指字段名，value1 指对应于字段 field01 的值。如想把 test1 表中王五的记录删除，可以执行以下操作。

```
delete from test1 where t name="王五"
```

正如 14.4.4 小节提到的那样，字段 ID 是表 test1 的主键，并且具有唯一性。所以如果把上面代码中的 name="王五"更改为 id=1 会更好的执行操作。下面就把这段代码加入到实际的 PHP 文件中，看如何使用 DELETE 语句来删除已经存在且不再需要的记录。具体内容请参看如下代码：

```
<html>
<head>
<title>删除表中记录</title>
</head>
<body>
<center>
<?php
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect(localhost,root,"")or die("不能连接到服务器".mysql_error());
mysql_select_db(test,$link);                    //选择相应的数据库，这里选择 test 库
function show_con()                             //把显示记录功能做成函数以便多次调用
{
    $link=mysql_connect(localhost,root,"");
    mysql_select_db(test);
    $sql="select * from test1";                 //先执行 SQL 语句显示所有记录以与更改后相比较
    $result=mysql_query($sql,$link);            //使用 mysql_query()发送 SQL 请求
    echo "<table border=1>";                    //使用表格格式化数据
    echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
    while($row=mysql_fetch_array($result))      //遍历 SQL 语句执行结果把值赋给数组
    {
        echo "<tr>";
        echo "<td>".$row[id]."</td>";           //显示 ID
        echo "<td>".$row[name]." </td>";        //显示姓名
        echo "<td>".$row[mail]." </td>";        //显示邮箱
        echo "<td>".$row[phone]." </td>";       //显示电话
        echo "<td>".$row[address]." </td>";     //显示地址
        echo "</tr>";
    }
```

```
        echo "</table>";
    }
    echo "删除前的记录为：";
    show_con();                              //调用函数，显示表中所有记录
    $sql="delete from test1 where id=1";     //创建更新记录 SQL 语句
    mysql_query($sql,$link);                 //发送 SQL 请求
    echo "<p>";
    echo "删除后的记录为：";
    show_con();                              //再次调用函数，显示表中记录以做比较
    ?>
    </center>
    </body>
    </html>
```

把以上代码保存为“14-17.php”。然后在 PHP 执行环境下运行该文件，执行结果将会如图 14.17 所示。

通过图 14.17 可以看出，在执行删除操作前，遍历表的结果，表中有两条记录。而在执行了删除操作后，ID 号为 1 的记录，已经从表中被删除了。说明删除操作顺利执行。



图 14.17　删除表中记录执行结果

14.4.7　执行 ALTER TABLE 语句改变表的结构

数据表在建立后，并不是一成不变的，有时需要增删表的某些字段。如通讯录表在建立后，只有姓名、邮箱、电话、地址几项可能还不能满足需要，还要再加上手机号一项新的内容。或者其中邮箱一项不再需要可予以删除。这时就要使用到 ALTER TABLE 语句来改变表的结构。该语句的使用格式如下所示：

```
alter table table_name add column field_name1 date,add column field_name2 time…;
```

其中的 table_name 是想要改变结构的表名，field_name1 与 field_name2 为想要添加的字段名，跟在它们后面的 date 和 time 表示该字段的类型。如要给表 test1 增加手机和生日列，就可以使用以下格式。

```
alter table test1 add column mob_phone varchar(11),add column birthday varchar(8);
```

可见要增加表的字段，后面要加上 add column 关键字。下面介绍如何删除某个字段。

删除字段也是使用 alter table 语句，只不过跟上 drop column 关键字。具体格式如下所示：

```
alter table table_name drop column field_name1,drop column field_name2…;
```

如要删除表 test1 中的邮箱一项，就可以使用以下格式。

```
alter table test1 drop mail;
```

通过对上面两种情况的介绍后，读者可以发现，无论是为表添加字段，还是删除表中已有字段，

均是使用 alter table 语句。不同的是增加字段后面跟的是 add column 关键字，而删除字段使用的是 drop column 关键字。

在实际执行这些操作之前，有必要先介绍一个与数据表列有关的函数 mysql_list_fields()。该函数的作用是返回指定表的所有字段。所以在变更表的结构之前先调用该函数，在执行完相关的操作后，再次调用该函数，就可以真实的对比执行前后的差别。该函数的使用格式如下所示：

```
mysql_list_fields(string database_name,string table_name[,resource link_identifier])
```

但是由于该函数已经过时，所以要使用 mysql_query()函数发送 SQL 请求“show columns from table_name”来替换该函数。

下面就实际地来使用 alter table 语句，看一看是如何对表的结构进行改变的。同时也来看一看如何使用相应的 SQL 语句“show columns from table_name”来返回表的所有字段。具体内容请参看以下代码：

```
<html>
<head>
<title>改变表的结构</title>
</head>
<body>
<center>
<?php
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect(localhost,root,"")or die("不能连接到服务器".mysql_error());
mysql_select_db(test,$link);                    //选择相应的数据库，这里选择 test 库
function show_field()                           //把显示字段功能做成函数以便多次调用
{
     $link=mysql_connect(localhost,root,"")or die("不能连接到服务器".mysql_error());
     mysql_select_db(test,$link);               //选择相应的数据库，这里选择 test 库
     $result=mysql_query("show columns from test1");
     echo "test1 表中现有的字段内容为：";
     while($row=mysql_fetch_array($result))     //遍历表中的所有字段
     {
          echo $row["Field"];
          echo "<p>";
     }
}
echo "删除前：";
echo "<p>";
show_field();                                   //显示表的所有字段
echo "<p>";
echo "删除后：";
echo "<p>";
$sql="alter table test1 add column mob_phone varchar(11),add column birthday varchar(8)";
mysql_query($sql);                              //执行添加字段的 SQL 语句
$sql="alter table test1 drop mail";
mysql_query($sql);                              //执行删除字段的 SQL 语句
show_field();                                   //再次显示表的所有字段
?>
```

```
</center>
</body>
</html>
```

把以上代码保存为“14-18.php”。然后在 PHP 执行环境中运行该文件，其执行结果如图 14.18 所示。



图 14.18　改变表的结构执行结果

从图 14.18 中可以看出，在执行改变表结构前表中有 5 个字段，分别是 id、name、mail、phone、address。可是在执行改变表的结构的操作后，字段变为了 6 个，分别是：id、name、phone、address、mob_phone、birthday。原来的 mail 字段被删除了，而又新添加了 mob_phone 和 birthday 两个字段。通过执行结果可以明显发现，改变表的结构的语句是如何发生作用。

随着对改变表的结构的学习，这一节对 MySQL 表进行操作也基本告一段落。简单回顾一下，这一节学习了如何插入记录、如何浏览记录、如何更新记录，如何改变表的结构等几部分的内容。这些内容是使用 MySQL 数据库时最常用到的。只有熟练掌握了这些基本内容，才能在这些基础上开发出基于 MySQL 数据库的高效率的 Web 应用程序。

14.5　对 MySQL 表的高级查询

14.4 节介绍的对数据表的浏览，只是简单的显示所有记录，并没有附加任何的条件，也没有对输出结果的约束。这一节介绍常用在 SELECT、DELETE 和 UPDATE 操作中所附加的子句。通过这些子句一方面能对执行的操作附加条件，而一方面也对显示的结果有所限制。这样就可以实现更多的功能了。

14.5.1　使用 WHERE 子句

不管是进行查询、删除或者修改操作时通常都是需要附加条件的。而 where 子句就为执行的 SELECT、DELETE 和 UPDATE 操作附加条件。如想要浏览表 test1 中 ID 号小于 10 的用户记录就可以这样使用 WHERE 子句：

```
select * from test1 where id<10
```

这样，只有满足条件（ID 小于 10 的记录）才会被浏览到。

同时 where 也可以用来作为数据搜索的条件。如果记录特别多，从中所有的记录中找出一条记录就非常麻烦，这时使用 where 子句就简单的多了。比如要查看某一个人（如：张三）的详细信息就可以这样使用 WHERE 子句：

```
select * from test1 where name='张三'
```

这样就可以显示出所有名字为张三的记录（可能不只一个人，因为可能有重名的情况）。

同时，WHERE 子句也可以用到 UPDATE 更新操作及 DELETE 删除操作中。

如在一个论坛中有很多用户，其中有普通会员、版主、超级版主、管理员等。到了节日时，要给所有的会员加分（实际上是改变积分的值），就可以根据用户不同的身份，给予不同的处理。如下面的几句。

下面这一句给普通用户增加 10 分：

```
update table_name set score=score+10 where shenfen="normal";
```

下面这一句给普通版主增加 100 分：

```
update table_name set score=score+100 where shenfen="banzhu";
```

下面这一句给超级版主增加 200 分：

```
update table_name set score=score+200 where shenfen="super";
```

经过以上 3 句 SQL 语句的操作，不同的用户就会被给予不同的积分奖励。

为什么非要用 3 句？用一句加上 3 个条件不是更好？注意这一点是不被允许的，因为 MySQL 并不支持多重 WHERE。所以以上 3 步操作不能放到一条语句中。

虽然 MySQL 并不支持多重 WHERE，但是可以使用 AND（和）、OR（或）这两个关键字，把多重条件组合起来使用。如要搜索 ID 号为 1 和 ID 号为 10 的用户的记录就可以使用：

```
select * from table_name where id=1 or id=10
```

这样就会查询出所需要的信息。同理 AND 操作是同时满足两个条件的内容才会被查询出。

WHERE 子句用于 DELETE 操作时是什么情况呢？如一个论坛中用户太多了。如果要删除连续一年没有登录的用户（一般是没有这样的操作的）。就可以这样使用：

```
delete from table_name where no_login>365;
```

这样，未登录时间超过一年的用户就会被删除。

还有一点需要注意，where 子句的操作符不仅有“=”、“<”、“>”等这些数学符号，还可以使用“not”表示逻辑非。如要搜索所有性别不为男的成员的信息就可以使用这样的格式：

```
select * from table_name where sex not male
```

这样就可以选出所有性别不是男的记录。

还有一个操作符 LIKE 可以用于模糊查询。它的使用格式如下所示：

```
select * from table_name where field_name like value
```

其中的 field_name 为字段（实际指代该字段的值），value 为某一个值。这样的使用是检查左边的字段的值 field_name 是否为右边的 value 的子串。

另外在使用 LIKE 操作符时还可以加上简单的通配符“%”、“_”。其中“%”指代 0 个或多个字符的字符串，“_”指代何一个单个字符。这样就可以通过 LIKE 的模糊查找功能找出包含某个字串的记录。如下面的操作：

```
select * from table_name where address like "%北京%"
```

这样就可以搜索出所有地址在北京的用户或者地址中包括有北京内容的用户的记录。

至此 WHERE 子句就为读者介绍到这里。可以看出该子句包含的内容还是相当广泛的。善于利用该子句可以大大的提高相关 SQL 语句的执行效率。

14.5.2　使用 LIMIT 子句对结果进行分页显示

在使用 SELECT 进行查询时，如果表中的结果太多，可能就会查找出很多的结果。如果只是想要其中的几条该怎么办。实际这种情况常出现在分页操作中。如一个留言簿一共有 100 条记录，如果只想让一页出现 10 条记录该怎么办？这时就要用到 LIMIT 子句。

LIMIT 跟 1 个或 2 个数字参数，如果给定 2 个参数，第 1 个指定要返回的第 1 行的偏移量，第 2 个指定返回行的最大数目。注意初始行的偏移量是 0（不是 1）。它的使用格式像下面这样：

```
select field_nam from table_name limit 10
```

下面使用一个实例来说明使用 limit 与不使用 limit 的区别（这里的表还使用 TEST1 表，在使用前使用前面的 14-10.php~14-12.php 给表中加入记录。不过表的结构在 14.4.7 小节中被改变。使用前还要再改回来）。

```
<html>
<head>
<title>使用 limit 子句</title>
</head>
<body>
<center>
<?php
$db_host=localhost;                      //MySQL 服务器名
$db_user=root;                           //MySQL 用户名
$db_pass="";                             //MySQL 用户对应密码
$db_name="test";                         //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);         //选择相应的数据库，这里选择 test 库
$sql="select * from test1";              //没有加入 limit 的 SQL 语句
$result=mysql_query($sql,$link);         //使用 mysql_query()发送 SQL 请求
echo "没有使用 limit 子句的输出结果：";
echo "<p>";
echo "<table border=1>";                 //使用表格格式化数据
echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
while($row=mysql_fetch_array($result))   //遍历 SQL 语句执行结果把值赋给数组
{
     echo "<tr>";
     echo "<td>".$row[id]."</td>";         //显示 ID
     echo "<td>".$row[name]." </td>";      //显示姓名
     echo "<td>".$row[mail]." </td>";      //显示邮箱
     echo "<td>".$row[phone]." </td>";     //显示电话
     echo "<td>".$row[address]." </td>";   //显示地址
```

```
        echo "</tr>";
    }
    echo "</table>";
    echo "<p>";
    echo "使用 limit 子句后的输出结果：";
    echo "<p>";
    $sql="select * from test1 limit 3";              //加入 limit 子句的 sql 语句
    $result=mysql_query($sql,$link);                  //使用 mysql_query()发送 SQL 请求
    echo "<table border=1>";                          //使用表格格式化数据
    echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
    while($row=mysql_fetch_array($result))            //遍历 SQL 语句执行结果把值赋给数组
    {
        echo "<tr>";
        echo "<td>".$row[id]."</td>";                 //显示 ID
        echo "<td>".$row[name]." </td>";              //显示姓名
        echo "<td>".$row[mail]." </td>";              //显示邮箱
        echo "<td>".$row[phone]." </td>";             //显示电话
        echo "<td>".$row[address]." </td>";           //显示地址
        echo "</tr>";
    }
    echo "</table>";
    ?>
    </center>
    </body>
    </html>
```

把以上代码保存为“14-19.php”。然后在 PHP 执行环境中运行该文件，执行结果如图 14.19 所示。

现在再对 14-19.php 的代码做小小的改动，把 limit 子句后面的参数 3 改为 23。即把原来的一个参数变为两个参数。第 1 个参数 2 为偏移量，第 2 参数 3 为显示的结果。然后再来执行 14-19.php，执行结果将会如图 14.20 所示。



ID	姓名	邮箱	电话	地址
2	李四	lisi@mircosoft.com	2345678	北京市海淀区
3	王五	wangwu@tom.com	1234567	某省市区街
4	赵六	zhaolin@126.com	1111111	北京市东城区
5	陈小二	xiaoer@sohu.com	2222222	河南省
6	赵匡胤	kuangyin@126.com	没有	北宋东京

ID	姓名	邮箱	电话	地址
2	李四	lisi@mircosoft.com	2345678	北京市海淀区
3	王五	wangwu@tom.com	1234567	某省市区街
4	赵六	zhaolin@126.com	1111111	北京市东城区

图 14.19　使用 limit 子句执行结果



ID	姓名	邮箱	电话	地址
2	李四	lisi@mircosoft.com	2345678	北京市海淀区
3	王五	wangwu@tom.com	1234567	某省市区街
4	赵六	zhaolin@126.com	1111111	北京市东城区
5	陈小二	xiaoer@sohu.com	2222222	河南省
6	赵匡胤	kuangyin@126.com	没有	北宋东京

ID	姓名	邮箱	电话	地址
4	赵六	zhaolin@126.com	1111111	北京市东城区
5	陈小二	xiaoer@sohu.com	2222222	河南省
6	赵匡胤	kuangyin@126.com	没有	北宋东京

图 14.20　使用 limit 子句加两个参数的执行结果

比较两次的执行结果可以明显看出用不用 LIMIT 子句的显示区别。不使用子句将显示所有记录，而使用子句后将显示指定条数的记录。加入一个参数只显示指定条数记录，加入两个参数把搜索结果按第 1 个参数进行偏移，再显示指定条数的记录。

14.5.3　使用 ORDER BY 对查询结果进行排序

使用 SELECT 语句对数据表进行浏览查询，所查出的记录都是按照记录插入的先后顺序排列的。有时这样的结果并不是用户所希望的。如果希望结果按照某一字段的某一顺序进行排列这时就需要用到 ORDER BY 子句。该子句的使用格式如下所示：

```
select field_nam from table_name order by field_name desc
```

其中的 field_name 为字段，table_name 为表名。参数 desc 为降序即从大到小的顺序排列，如果要使用升序即从大到小的顺序排列就要使用参数 asc。如果不使用参数，默认为按照升序进行排序。如对表 test1 中的记录按电话号码一项按降序排列就可以这样使用：

```
select * from test1 order by phone desc;
```

下面通过一个实例来看一下如何使用 order by 子句对查询结果进行排序。具体内容参看以下代码：

```
<html>
<head>
<title>使用 order by 子句</title>
</head>
<body>
<center>
<?php
$db_host=localhost;                          //MySQL 服务器名
$db_user=root;                               //MySQL 用户名
$db_pass="";                                 //MySQL 用户对应密码
$db_name="test";                             //要操作的数据库
//使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);             //选择相应的数据库，这里选择 test 库
$sql="select * from test1";                  //没有加入 limit 的 SQL 语句
$result=mysql_query($sql,$link);             //使用 mysql_query()发送 SQL 请求
echo "没有使用 order by 子句的输出结果：";
echo "<p>";
echo "<table border=1>";                     //使用表格格式化数据
echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
while($row=mysql_fetch_array($result))       //遍历 SQL 语句执行结果把值赋给数组
{
    echo "<tr>";
    echo "<td>".$row[id]."</td>";            //显示 ID
    echo "<td>".$row[name]." </td>";         //显示姓名
    echo "<td>".$row[mail]." </td>";         //显示邮箱
    echo "<td>".$row[phone]." </td>";        //显示电话
    echo "<td>".$row[address]." </td>";      //显示地址
```

```
        echo "</tr>";
}
echo "</table>";
echo "<p>";
echo "使用 order by 子句后的输出结果：";
echo "<p>";
$sql="select * from test1 order by phone";          //加入 order by 子句的 sql 语句
$result=mysql_query($sql,$link);                    //使用 mysql_query()发送 SQL 请求
echo "<table border=1>";                            //使用表格格式化数据
echo "<tr><td>ID</td><td>姓名</td><td>邮箱</td><td>电话</td><td>地址</td></tr>";
while($row=mysql_fetch_array($result))              //遍历 SQL 语句执行结果把值赋给数组
{
        echo "<tr>";
        echo "<td>".$row[id]."</td>";               //显示 ID
        echo "<td>".$row[name]." </td>";            //显示姓名
        echo "<td>".$row[mail]." </td>";            //显示邮箱
        echo "<td>".$row[phone]." </td>";           //显示电话
        echo "<td>".$row[address]." </td>";         //显示地址
        echo "</tr>";
}
echo "</table>";
?>
</center>
</body>
</html>
```

把以上代码保存为“14-20.php”。然后在 PHP 执行环境中运行该文件，执行结果如图 14.21 所示。查看图 14.21 的执行结果，可以看出，下部的输出结果是按照所有记录的电话升序进行排序。

现在把上例中的 ORDER BY 子句后面加上参数 desc。再次执行，结果会如图 14.22 所示。



图 14.21　使用 order by 子句执行结果

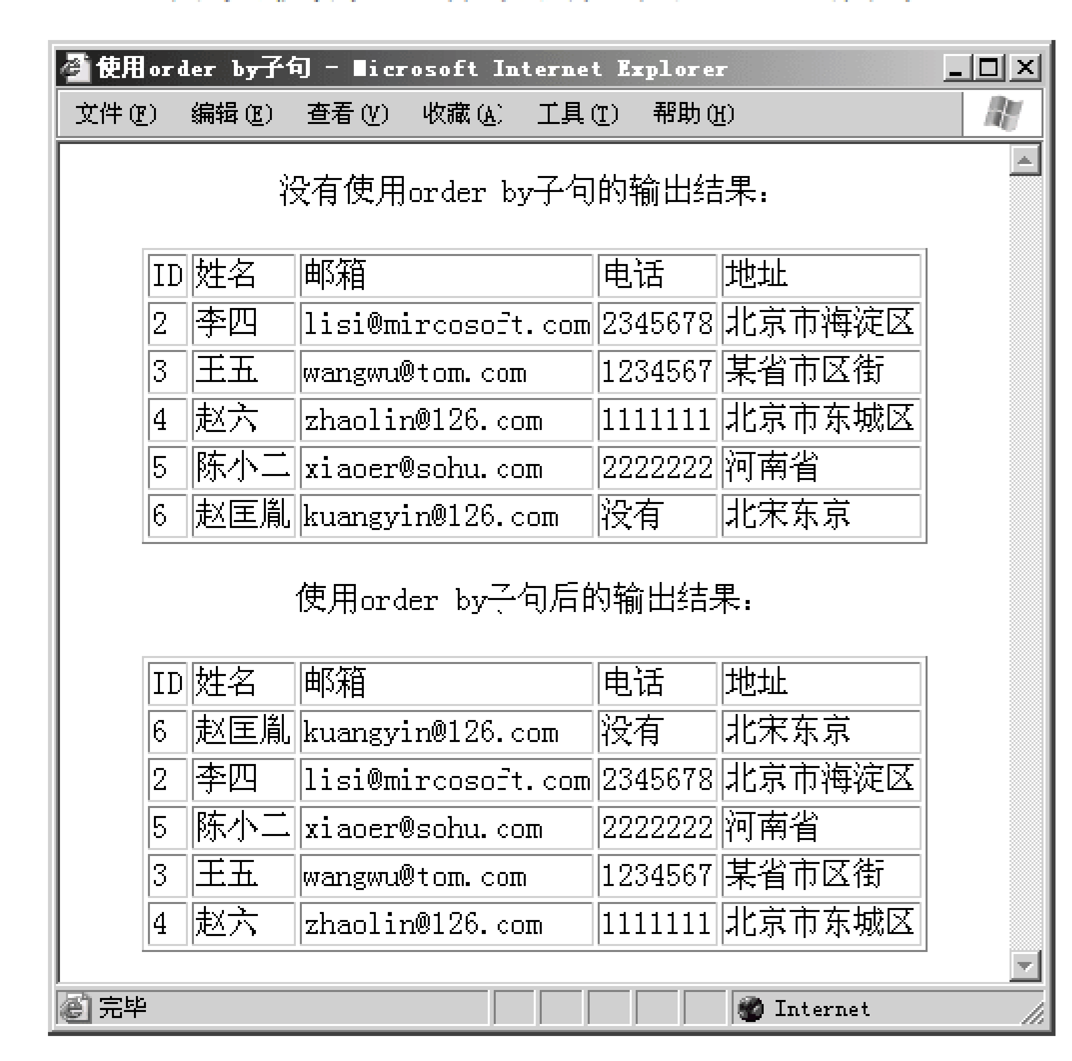


图 14.22　加上 desc 参数后的执行结果

而图 14.22 的执行结果则显示，下部的输出按所有记录的电话进行降序排列。

从以上两次执行结果图中可以清晰的看出，使用 ORDER BY 子句与不使用该子句的区别。

另外在使用 ORDER BY 子句时有一项需要注意，ORDER DY 子句允许多次使用。这样就可以实现多重排序。就像下面这样：

```
select field_nam from table_name order by field_name,order by field_name2
```

如何理解呢？就是说，先对查询结果按照 field_name 字段按升序进行排列，如果有多个记录它们的 field_name 字段都相同，这时再按照 field_name2 字段按升序进行排列。这就是多重排序。比如说对某个班级的学生成绩先按照总成绩进行排序，以排出名次。如果两个学生的总成绩都相同，再按照语文的成绩进行排序。

当然也可以使用 3 种甚至多种排序。即 ORDER BY 子句可以多次使用。

高级查询的内容，到这里也就为读者介绍完了。利用本节介绍的几种高级查询子句一是可以提高 SQL 语句执行效率，另外还可以结合不同的功能扩展 SQL 语句的功能。

14.6　MySQL 数据库使用实例

14.6.1　明确设计目的——学生档案管理

这一节，结合本章所学 MySQL 数据库的内容，来做一个实际使用的例子。制作一个学生档案管理系统。

一个班级学生档案管理系统必备的功能就是对学生信息的管理，包括添加信息、浏览信息、修改信息、删除信息等操作。

这里先建立一个表 student 以存放学生的信息。该表中有以下字段，ID（记录号）、NAME（姓名）、SEX（性别）、BIRTHDAY（出生日期）、S_ID（学号）、PARENT（家长姓名）、PHONE（家庭电话）、ADDRESS（家庭地址）等。

把一些变量存为一个配置文件中，以便多次调用，该文件命名为 14-21.php。其内容如下所示：

```
<?php
$host_name="localhost";                    //主机名
$db_user="root";                           //用户名
$db_pass="";                               //用户密码
$db_name="test";                           //数据库名
$table_name="student";                     //数据表名
$list_num=10;                              //每页显示记录数
?>
```

该文件存放了连接数据库时必须的各项内容，这样需要时只需使用 include()或者 require 调用该文件即可，避免了多次定义。

准备工作这一小节就介绍到这里。从 14.6.2 节开始具体的设计工作。

14.6.2　连接 MySQL 服务器建立学生档案表

这一节来介绍连接到服务器创建学生档案表 student。具体内容参看以下代码：

```
<?
require "14-21.php";                              //调用配置文件
//连接 MySQL 服务器
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);                  //选择 test 数据库
//下面的$sql 就为创建表的 SQL 语句
$sql="create table $table_name(
    id int(5) not null auto_increment primary key,
    name varchar(12) not null,
    sex varchar(4) not null,
    birthday varchar(16) not null,
    s_id varchar(10) not null,
    parent varchar(12) not null,
    phone varchar(14) not null,
    address varchar(30) not null
)";
if(mysql_query($sql,$link))                       //发送 SQL 语句执行创建表的操作
echo "表".$table_name."创建成功";                  //如果创建成功显示信息
else
echo "创建数据库出错！";
?>
```

把以上代码保存为“14-22.php”。然后在 PHP 执行环境中运行该文件，将会出现如图 14.23 所示的执行结果。

创建表是建立系统的基础，其他一切操作都要在表中进行。表创建完成后，下一步就是为表中添加相应的数据了。14.6.3 小节将介绍显示数据页面及添加数据页面的创建过程。



图 14.23　建立学生档案表执行结果

14.6.3　显示学生数据页的创建

这一小节为读者介绍如何显示表中的数据。

显示数据当然还是要使用 SELECT 查询，但这一小节的 SELECT 查询不同于前面介绍的内容。这一节所使用的 SELECT 查询包括更多的内容，最重要的一项就是使用 LIMIT 子语句以实现分页功能。

```
<?
require "14-21.php";                              //调用配置文件
if(!$_GET[page])
$page=1;
else
$page=$_GET[page];
```

```
//连接 MySQL 服务器
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);                    //选择 test 数据库
//下面的$sql 就为创建表的 SQL 语句
$sql="select id from $table_name";                  //查询所有记录
$result=mysql_query($sql,$link);                    //发送 SQL 请求
$num=mysql_num_rows($result);                       //获得记录数
?>
<html>
<head>
<title>学生档案管理系统——记录显示页</title>
</head.
<body>
<center>
<h1>学生档案管理系统——记录显示页</h1>
<p>
<a href=14-24.php>添加记录</a>   <a href=14-26.php>查找记录</a>
<p>
<?
echo "目前共有".$num."条记录  ";          //输出记录数
$p_count=ceil($num/$list_num);                      //总页数为总条数除以每页显示数
echo "共分".$p_count."页显示  ";          //输出页数
echo "当前显示第".$page."页";
echo "<p>";
if($num>0)                                          //如果记录数大于 0 输出记录内容
{
?>
<p>
<table border="1">
<tr>
<td>ID</td>
<td>姓名</td>
<td>性别</td>
<td>出生日期</td>
<td>学号</td>
<td>家长姓名</td>
<td>家庭电话</td>
<td>家庭住址</td>
<td> </td>
</tr>
<?
$temp=($page-1)*$list_num;
$sql="select * from $table_name limit $temp,$list_num";
$result=mysql_query($sql);                          //执行 SQL 语句
while($row=mysql_fetch_array($result))              //通过循环遍历记录集
{
    echo "<tr>\n";
    echo "<td>".$row[id]."</td>\n";
    echo "<td>".$row[name]."</td>\n";
```

```
        echo "<td>".$row[sex]."</td>\n";
        echo "<td>".$row[birthday]."</td>\n";
        echo "<td>".$row[s_id]."</td>\n";
        echo "<td>".$row[parent]."</td>\n";
        echo "<td>".$row[phone]."</td>\n";
        echo "<td>".$row[address]."</td>\n";
        echo "<td><a href=14-25.php?id=".$row[id].">改</a></td>\n";
        echo "</tr>\n";
    }
    echo "</table>";
    //以下为显示分页的连接的内容
    $prev_page=$page-1;                         //定义上一页为该页减 1
    $next_page=$page+1;                         //定义下一页为该页加 1
    echo "<p align=\"center\"> ";
    if ($page<=1)                               //如果当前页小于等于 1 只有显示
    {
        echo "第一页 | ";
    }
    else                                        //如果当前页大于 1 显示指向第一页的连接
    {
        echo "<a href='$_SERVER[PHP_SELF]?page=1'>第一页</a> | ";
    }
    if ($prev_page<1)                           //如果上一页小于 1 只显示文字
    {
        echo "上一页 | ";
    }
    else                                        //如果大于 1 显示指向上一页的连接
    {
        echo "<a href='$_SERVER[PHP_SELF]?page=$prev_page'>上一页</a> | ";
    }
    if ($next_page>$p_count)                    //如果下一页大于总页数只显示文字
    {
        echo "下一页 | ";
    }
    else                                        //如果小于总页数则显示指向下一页的连接
    {
        echo "<a href='$_SERVER[PHP_SELF]?page=$next_page'>下一页</a> | ";
    }
    if ($page>=$p_count)                        //如果当前页大于或者等于总页数只显示文字
    {
        echo "最后一页</p>\n";
    }
    else                                        //如果当前页小于总页数显示最后页的连接
    {
        echo "<a href='$_SERVER[PHP_SELF]?page=$p_count'>最后一页</a></p>\n";
    }}
    else                                        //如果没有记录时输出信息
    {
        echo "暂时还没有记录！";
```

```
}
?>
</body>
</html>
```

把以上代码保存为“14-23.php”，以备后用。

可以看出，14-23.php 中的 SELECT 与前几节介绍的略有不同，不同之处就在于它使用了 LIMIT 的两个参数。第 1 个参数为返回查询的偏移量，第 2 个参数为显示的记录数。记录数很简单，直接限制为每页最多显示数就可以了。而偏移量怎么计算呢？如当前页为第 1 页，偏移量为 0 就可以了；当前页为第 2 页，偏移量就为 10；当前页为第 3 页，偏移量就为 20……以此类推。如果当前页为 N，偏移量就为（（N-1）*$list_num）即前面页面显示的所有记录数，这样就实现了分页的显示。

14.6.4　添加记录页的创建

由于表中还没有记录，所以在执行 14-23.php 时会得到没有记录的结果。这一节就开始着手来为表中添加记录。这里还是采用前台输入加后台处理的方式来进行。不过这一小节，把这前台的表单输入和后台的数据处理整合到一个文件中，以使程序更加简洁。具体内容请参看以下代码：

```
<?
if(!$_POST[name])                                    //如果没有记录输入
{
?>
<html>
<head>
<title>学生档案管理系统——记录添加页</title>
</head.
<body>
<script language="javascript">
function Juge(theForm)
{
	if (theForm.name.value == "")
	{
		alert("请输入姓名！");
		theForm.name.focus();
		return (false);
	}
	if (theForm.s_id.value == "")
	{
		alert("请输入学号！");
		theForm.s_id.focus();
		return (false);
	}
}
</script>
<center>
<h1>学生档案管理系统——记录添加页</h1>
```

```
<a href="14-23.php">返回首页</a>
<table border=1>
<form action="<?echo $PATH_INFO?>" method="post" onsubmit="return Juge(this)">
<tr>
<td>输入姓名：</td>
<td><input name="name" type="text"></td>
</tr>
<tr>
<td>输入性别：</td>
<td>
<input type=radio name=sex value=男  checked>男
<input type=radio name=sex value=女>女
</td>
</tr>
<tr>
<td>出生日期：</td>
<td>
<select name=b_y>
<?
for($i=1980;$i<2004;$i++)                              //循环输出出生年
echo "<option value=".$i.">".$i."\n";
?>
</select>年
<select name=b_m>
<?
for($i=1;$i<13;$i++)                                   //循环输出出生月
echo "<option value=".$i.">".$i."\n";
?>
</select>月
<select name=b_d>
<?
for($i=1;$i<32;$i++)                                   //循环输出出生日
echo "<option value=".$i.">".$i."\n";
?>
</select>日
</td>
</tr>
<tr>
<td>输入学号：</td>
<td><input name="s_id" type="text"></td>
</tr>
<tr>
<td>监护人姓名：</td>
<td><input name="parent" type="text"></td>
</tr>
<tr>
<td>家庭电话：</td>
<td><input name="phone" type="text"></td>
</tr>
```

```
<tr>
<td>家庭住址：</td>
<td><input name="address" type="text"></td>
</tr>
<tr>
<td colspan="2"><center><input type=submit value="确认提交">
<input type=reset value="重新选择"></center></td>
</tr>
</form>
</table>
<?
}
else
{
//以下内容为获取表单传递的变量
    $name=$_POST[name];
    $sex=$_POST[sex];
    $birthday=$_POST[b_y]."年".$_POST[b_m]."月".$_POST[b_d]."日";
    $s_id=$_POST[s_id];
    $parent=$_POST[parent];
    $phone=$_POST[phone];
    $address=$_POST[address];
    require "14-21.php";
    $link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
    mysql_select_db($db_name,$link);                    //选择 test 数据库
    $sql="insert        into        $table_name        (name,sex,birthday,s_id,parent,phone,address)
values('$name','$sex','$birthday','$s_id','$parent','$phone','$address')";
    mysql_query($sql,$link);                            //执行插入记录的 SQL 语句
?>
<html>
<head>
<title>学生档案管理系统——记录添加页</title>
</head>
<meta http-equiv="refresh" content="2; url=14-23.php">
<body>
已经成功添加记录，两秒后返回。
</body>
</html>
<?
}
?>
```

把以上代码保存为“14-24.php”，以备后用。

可以看出该文件与前面对表进行操作时的例子有所不同。不同之处就在于以前是输入数据的前台与处理数据的后台是分开的，是不同的两个文件。现在把前台与后台的功能整合到一个文件中了。其中的关键就是表单的 action 属性要指向自身。这里使用了 PHP 的系统变量$PATH_INFO 即指代文件自身。同时还用 POST 变量来判断前后台操作。如果没有使用 POST 变量则判定为前台添加数据操作，如果有了 POST 变量则判断为后台处理添加数据的操作。

14.6.5　修改记录页的创建

文件 14-23.php 中有一个指向 14-25.php 的连接，该文件的作用就是对原有的数据进行修改。这一节就来创建这个修改原有记录的文件。同 14.6.4 节一样，这个文件仍然把前台输入修改内容与后台处理修改结果整合为一个文件。具体内容参看以下代码：

```
<?
if(!$_POST[name])                                    //如果没有记录输入
{
?>
<html>
<head>
<title>学生档案管理系统——记录修改页</title>
</head.
<body>
<script language="javascript">
function Juge(theForm)
{
	if (theForm.name.value == "")
	{
		alert("请输入姓名！");
		theForm.name.focus();
		return (false);
	}
	if (theForm.s_id.value == "")
	{
		alert("请输入学号！");
		theForm.s_id.focus();
		return (false);
	}
}
</script>
<center>
<h1>学生档案管理系统——记录修改页</h1>
<a href="14-23.php">返回首页</a>
<table border=1>
<form action="<?echo $PATH_INFO?>" method="post" onsubmit="return Juge(this)">
<?
if(!$_GET[id])
{
	echo "没有选择 ID 值";
	exit();
}
else
{
	$id=$_GET[id];
}
```

```
require "14-21.php";
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);                    //选择 test 数据库
$sql="select * from $table_name where id='$id'";    //创建 SQL 语句
$result=mysql_query($sql,$link);                    //执行 SQL 查询
$row=mysql_fetch_array($result);                    //获取数组
$temp=$row[birthday];                               //把出生日期赋值给变量
//以下代码为通过不同情况分离出出生的年月日
$b_y=substr($temp,0,4);                             //从出生日期中提取年
if(strrpos($temp,"月")=="7")                        //查找月的位置以判断月份的位数
{
    $b_m=substr($temp,6,1);                         //从出生日期中提取出月
    if(strrpos($temp,"日")=="10")                   //查找日的位置以判断日期的位数
    {
        $b_d=substr($temp,9,1);                     //从出生日期中提取出日
    }
    else
    $b_d=substr($temp,9,2);                         //从出生日期中提取出日
}
else
{
    $b_m=substr($temp,6,2);
    if(strrpos($temp,"日")=="11")
    {
        $b_d=substr($temp,10,1);
    }
    else
    $b_d=substr($temp,10,2);
}
?>
<tr>
<td>输入姓名：</td>
<td><input name="name" type="text" value="<?echo $row[name]?>"></td>
</tr>
<tr>
<td>输入性别：</td>
<td>
<input type=radio name=sex value=男  <? if ($row[sex]=="男") echo "checked"?>>男
<input type=radio name=sex value=女  <? if ($row[sex]=="女") echo "checked"?>>女
</td>
</tr>
<tr>
<td>出生日期：</td>
<td>
<select name=b_y>
<?
for($i=1980;$i<2004;$i++)                           //循环输出出生年
{
    echo "<option value=".$i;
```

```
        if($b_y==$i) echo " selected=1";
        echo ">".$i."\n";
}
?>
</select>年
<select name=b_m>
<?
for($i=1;$i<14;$i++)                                    //循环输出出生月
{
        echo "<option value=".$i;
        if($b_m==$i) echo " selected=1";
        echo ">".$i."\n";
}
?>
</select>月
<select name=b_d>
<?
for($i=1;$i<32;$i++)                                    //循环输出出生日
{
        echo "<option value=".$i;
        if($b_d==$i) echo " selected=1";
        echo ">".$i."\n";
}
?>
</select>日
</td>
</tr>
<tr>
<td>输入学号：</td>
<td><input name="s_id" type="text" value="<?echo $row[s_id]?>"></td>
</tr>
<tr>
<td>监护人姓名：</td>
<td><input name="parent" type="text" value="<?echo $row[parent]?>"></td>
</tr>
<tr>
<td>家庭电话：</td>
<td><input name="phone" type="text" value="<?echo $row[phone]?>"></td>
</tr>
<tr>
<td>家庭住址：</td>
<td><input name="address" type="text" value="<?echo $row[address]?>"></td>
</tr>
<tr>
<td colspan="2"><center><input type=submit value="确认提交">
<input type=reset value="重新选择"></center></td>
</tr>
</form>
</table>
```

```
<?
}
else
{
//以下内容为获取表单传递的变量
	$name=$_POST[name];
	$sex=$_POST[sex];
	$birthday=$_POST[b_y]."年".$_POST[b_m]."月".$_POST[b_d]."日";
	$s_id=$_POST[s_id];
	$parent=$_POST[parent];
	$phone=$_POST[phone];
	$address=$_POST[address];
	require "14-21.php";
	$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
	mysql_select_db($db_name,$link);				//选择 test 数据库
	$sql="update				$table_name				set
name='$name',sex='$sex',birthday='$birthday',s_id='$s_id',parent='$parent',phone='$phone',address='$address'";
	mysql_query($sql,$link);				//执行更新记录的 SQL 语句
?>
<html>
<head>
<title>学生档案管理系统——记录添加页</title>
</head>
<meta http-equiv="refresh" content="2; url=14-23.php">
<body>
已经成功更改记录，两秒后返回。
</body>
</html>
<?
}
?>
```

把以上代码保存为“14-25.php”，以备后用。

该文件的一个特征是把修改记录的前台输入内容与后台处理数据结合了起来。但这里还存在一个问题。在输入学生的出生日期时，把输入记录格式化后存入表中，当需要修改时，如果要求显示原来的出生日期，就必须把原来的数据还原。这里用了比较笨一点的方法按位数分别提取出年、月、日来实现。其实如果在创建表时把出生年、月、日分别创建成字段就不会有这样的问题存在了。

14.6.6　查找记录页的创建

如果班级学生的人数特别多，记录分了好多页显示。要从表中查找某一条或几条特定的记录是相当困难的。所以，有必要创建一个查找记录的页面。这一节就来设计该页面。该页面首先让用户选择查找记录所依据的条件，然后，输入该条件的内容。如用户先选择学号，再输入学号的内容如“200602003”，然后就把结果交给后台来处理了。同 14.6.4 节和 14.6.5 节一样，查找记录的页面，依然采用前台与后台整合在一起的模式。具体内容请参见以下代码：

```
<html>
<head>
<title>学生档案管理系统——记录查找页</title>
</head>
<body >
<center>
<h1>学生档案管理系统——记录查找</h1>
<p>
<a href=14-23.php>返回</a>
<p>
<?
if(!$_POST["find_v"])                        //如果没有查找类别，则显示 HTML 内容
{
?>
<script language="javascript">
function Juge(theForm)
{
    if (theForm.find_v.value == "")
    {
        alert("请输入查询内容！");
        theForm.find_v.focus();
        return (false);
    }
}
</script>
<table border="1">
<form action=<?echo $PATH_INFO?> method="post" onsubmit="return Juge(this)">
<tr>
<td>选择查询类别：</td>
<td><select name="find_t">
<option value=name>姓名</option>
<option value=s_id>学号</option>
<option value=birthday>出生日期</option>
<option value=parent>监护人</option>
<option value=phone>电话</option>
<option value=address>住址</option>
</select></td>
</tr>
<tr>
<td>输入查询内容：</td>
<td><input type=text name="find_v"></td>
</tr>
<tr>
<td colspan=2><center><input type=submit value=确认提交>
<input type=reset value=重新填写></center></td>
</tr>
</table>
</center>
<?
```

```
}
else
{
require "14-21.php";                                    //调用配置文件
$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
mysql_select_db($db_name,$link);                        //选择 test 数据库
$strsql="select*from $table_name where $_POST[find_t] like '$_POST[find_v]'";
$result=mysql_query($strsql,$link) or die(mysql_error());
$amount=mysql_num_rows($result);
if($amount=="0") echo "<p>没有记录";                    //如果没有找到匹配记录，则输出提示
else
{
?>
<h3>下面是查出来的结果</h3>
<p>
<table border=1>
<tr>
<td>ID</td>
<td>姓名</td>
<td>性别</td>
<td>出年日期</td>
<td>学号</td>
<td>监护人</td>
<td>电话</td>
<td>地址</td>
</tr>
<?
    while($row=mysql_fetch_array($result))
    {
        echo "<tr>
        <td>".$row["id"]."</td>
        <td>".$row["name"]."</td>
        <td>".$row["sex"]."</td>
        <td>".$row["birthday"]."</td>
        <td>".$row["s_id"]."</td>
        <td>".$row["parent"]."</td>
        <td>".$row["phone"]."</td>
        <td>".$row["address"]."</td>
        </tr>";
    }
    echo "</table></center>";
}
}
?>
</body>
</html>
```

把以上代码保存为“14-26.php”，以备后用。

该文件有两个要点。一个是使用了 WHERE 子句，另外一个是使用了 WHERE 子句的 LIKE 操作

符。即带条件的模糊查找。这样虽然降低了查找精度，但却提高了查找的准确性。

随着该文件的创建完毕，该学生档案管理系统，也基本上成型了。下面就来实际地测试一下该系统的使用情况，以排除 BUG。

14.6.7　学生档案管理系统实际使用

由于表已经在 14.6.2 小节创建完毕，所以，直接从 14-23.php 开始执行。由于表中并没有任何记录，所以是首次执行该系统，其执行结果将会如图 14.24 所示。

（1）单击“添加记录”链接后，执行 14-24.php，出现如图 14.25 所示的执行结果。

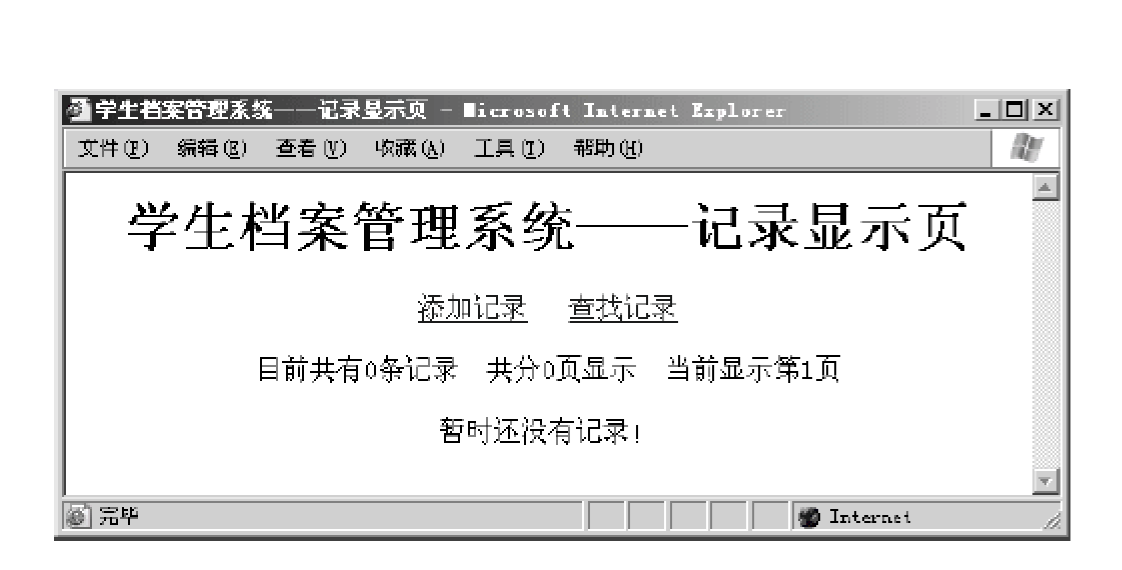


图 14.24　记录显示页第 1 次执行结果

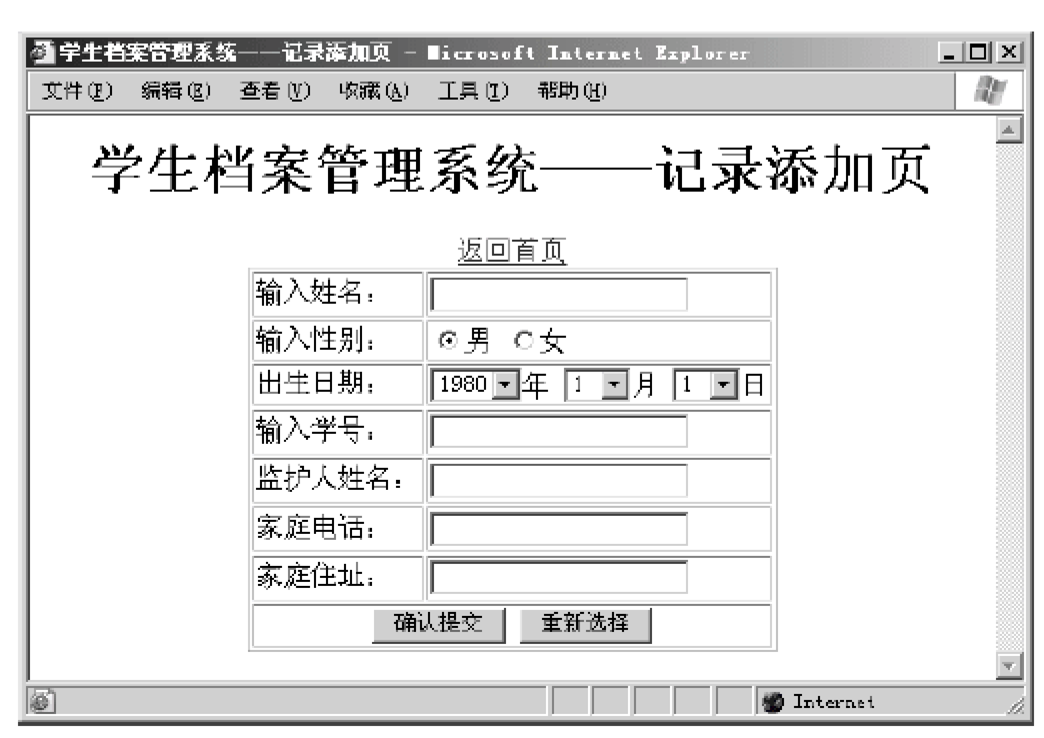


图 14.25　执行记录添加页执行结果

（2）这里，按要求填入适当的学生信息。如比利、男、1999 年 5 月 20 日、200601001、大比利、1234567、东区十三街 28 号。输入内容后，单击“确认提交”按钮。由于系统在处理完提交的记录后，就会直接转到记录显示页。所以提交过后的记录显示页如图 14.26 所示。可以看出，已经显示出了所添加的记录，说明添加记录及显示记录均执行正常。

（3）这时就可以对记录 1 进行修改了。例如，记错了门牌号，把原本 26 号写成了 28 号，那么现在把门牌号改过来。单击记录后面的“改”链接，将执行 14-25.php。其结果如图 14.27 所示。

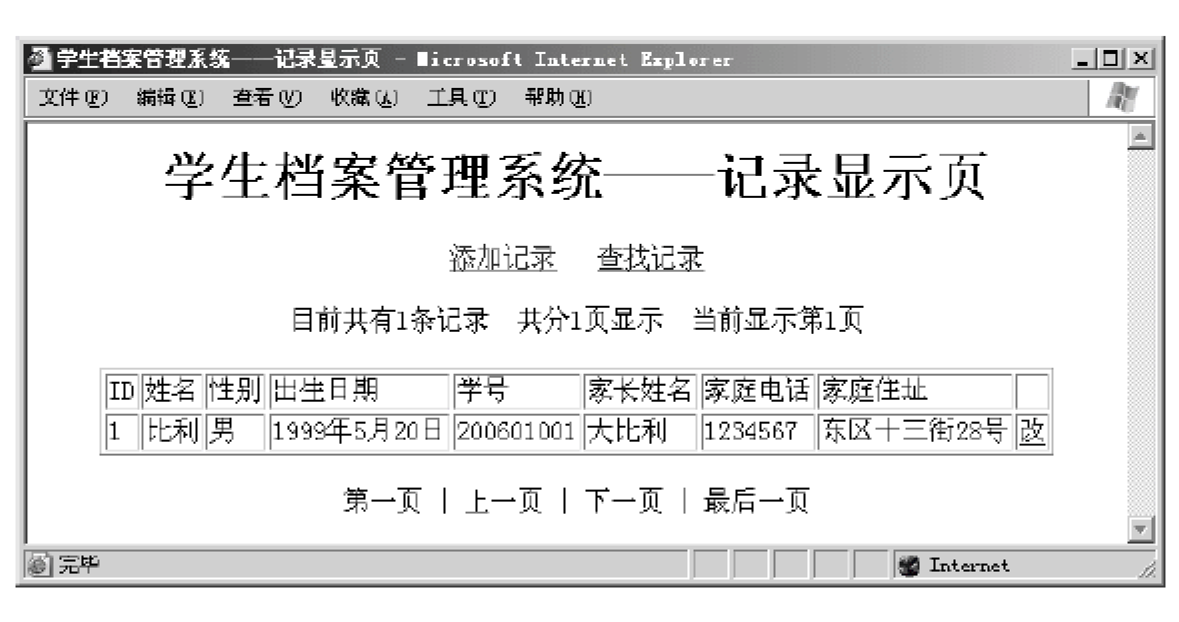


图 14.26　已经存在记录的记录显示页执行结果

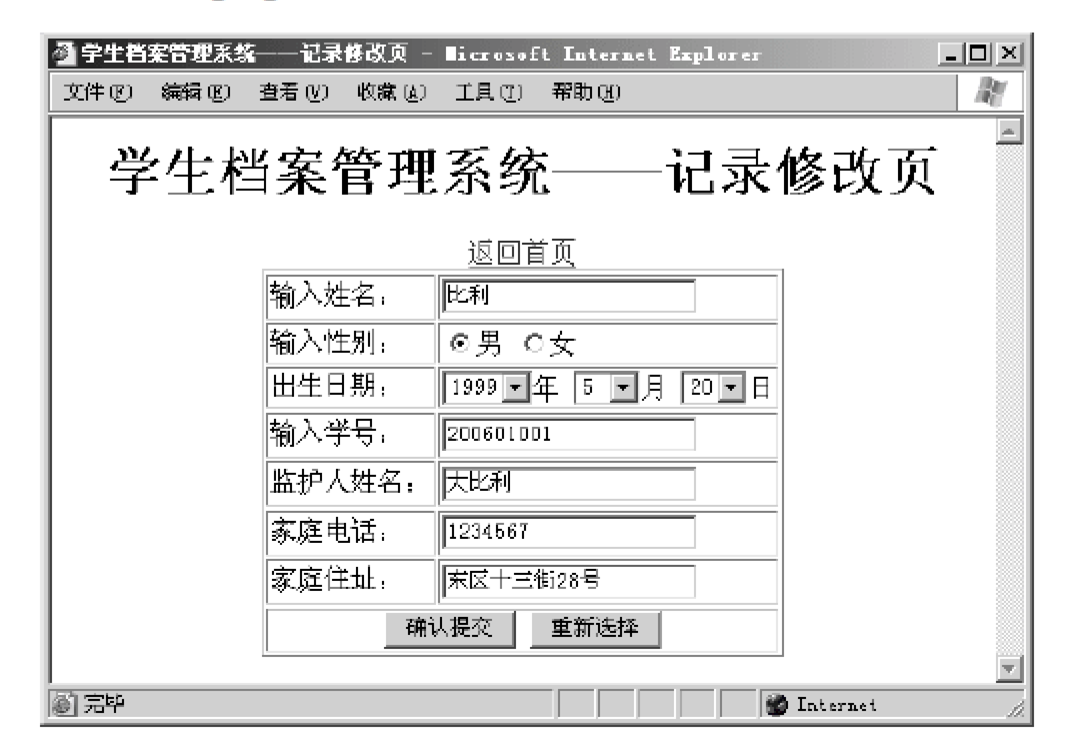


图 14.27　记录修改页执行结果

（4）从图 14.27 可以看出，该页正确显示出了原有的记录内容，并等待用户输入新的信息。这里

把家庭住址一项的“东区十三街 28 号”其中的 28 号改为 26 号。然后单击“确认提交”按钮，修改后，直接跳转到了记录显示页。结果如图 14.28 所示。

从图 14.28 的执行结果可以看出家庭住址一项已经由原来的“东区十三街 28 号”变成了“东区十三街 26 号”。说明修改记录页也执行正常。

下面测试记录显示页的分页功能。要想测试此功能的前提是表中存放的记录数要多于在 14-21.php 中所定义的$list_num 每页显示数。这里继续为表中添加记录，以测试分页显示的功能。

（1）为了方便查看分页功能，暂时把 14-21.php 中的$list_num 改为“5”。当表中记录多于 5 条时，14-23.php 的执行结果将如图 14.29 所示。

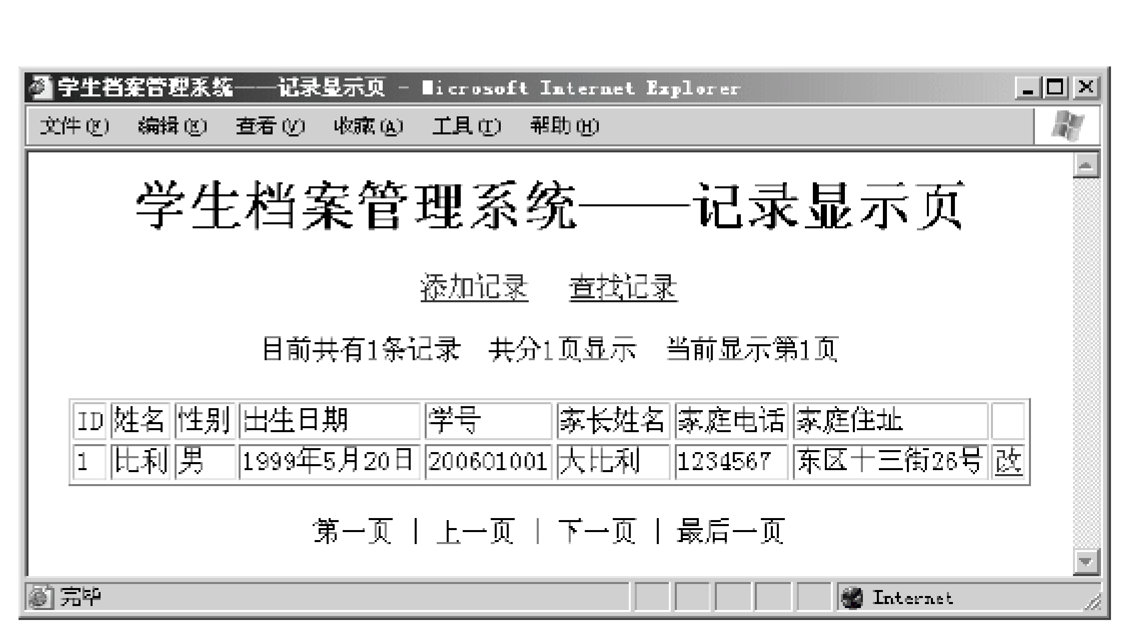


图 14.28　记录修改后记录显示页执行结果

图 14.29　记录多于一页的执行结果

（2）从图 14.29 可以看出，这时已经显示出了指向下一页的链接。单击该链接将显示后面的两条记录，如图 14.30 所示。从图 14.29 及图 14.30 可以发现，分页功能已经得到实现。

下面继续来测试查找记录的功能。

（1）在主页面上单击“查找记录”的链接，将会转到查找记录页（执行 14-26.php），执行结果如图 14.31 所示。

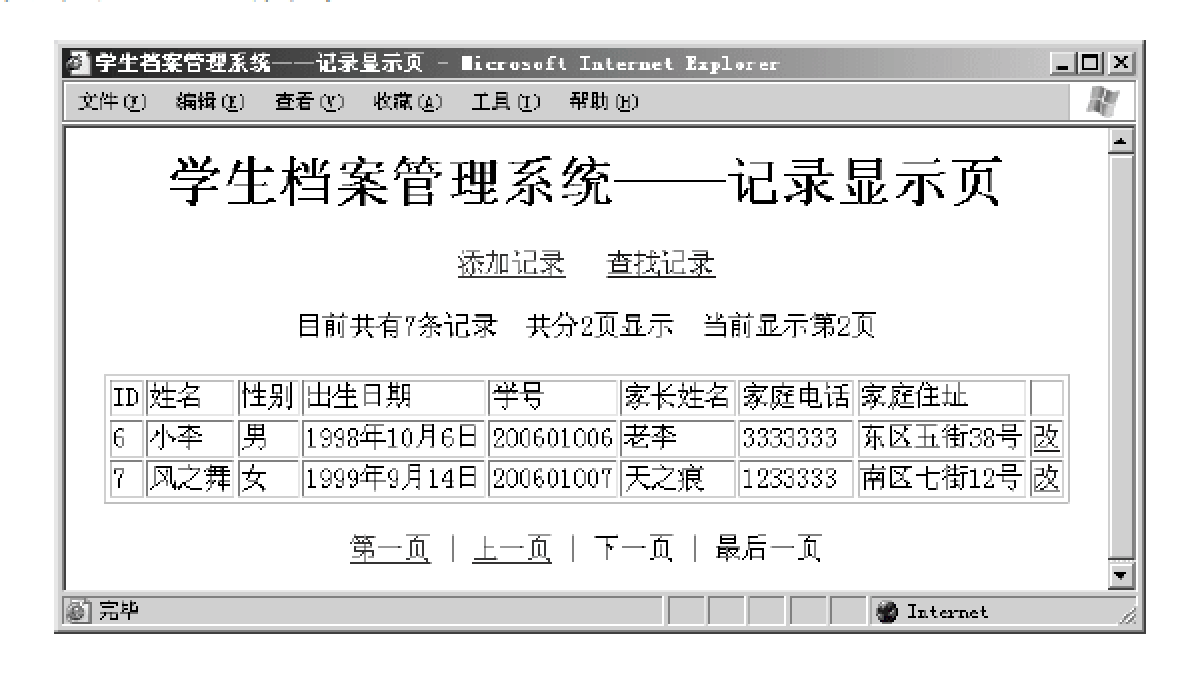


图 14.30　打开最后一页链接的结果

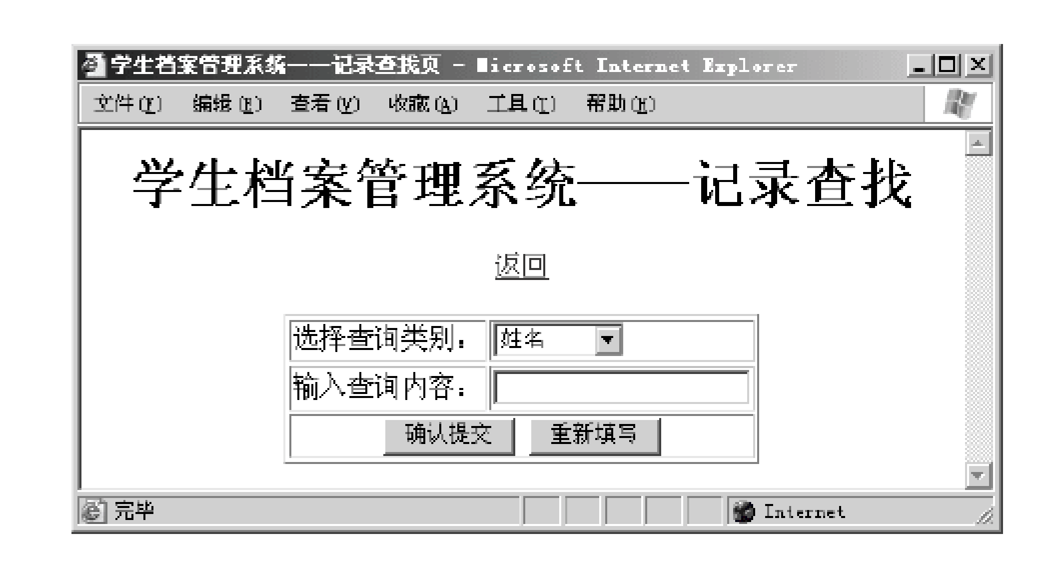


图 14.31　查找记录前台输入执行结果

（2）这里，选择姓名作为选择条件，在查询内容一栏中输入“小李”，然后单击“确认提交”按钮，将出现如图 14.32 所示的执行结果。

结果显示出了姓名为“小李”的学生的记录。说明记录查找功能正常运行。

至此，该系统基本上已经测试完毕。

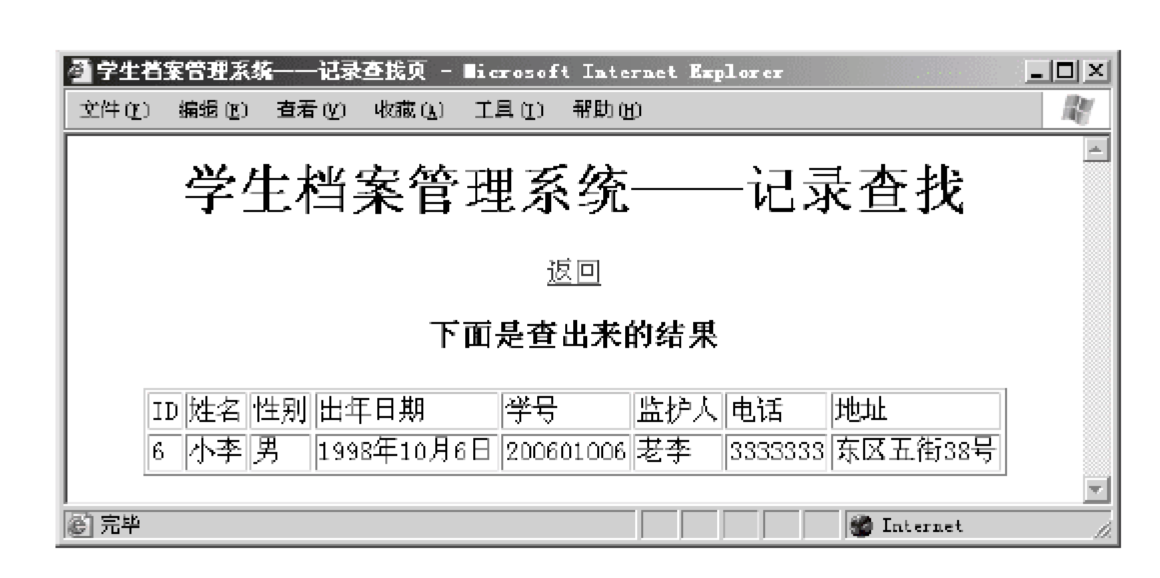


图 14.32　查找记录，查找结果

不过系统还有个模块，如学生转学，就不用继续保留其记录。就要将相应的记录从库表中删除，即记录的删除的模块。结合前面所介绍的内容，这个功能也是很容易实现的。有兴趣的读者不妨自己动手把删除记录这个功能添加上。这样整个系统就更加完善了。

这一节，通过一个简单的实例，读者可把常用的数据库表操作组织起来，可以说是对学过知识的一次检阅。其实说到底，数据库操作最常用的也就是浏览、添加、修改、删除等。至于用何种方式，如何去实现这些功能，就需要在实践中不断摸索了。

14.7　本 章 小 结

数据库应用是动态网页技术最重要的应用之一。动态网页技术只有充分与数据库结合，才能发挥动态网页技术的真正魅力。PHP 与 MySQL 是一对黄金组合，所以学习 PHP 自然就要学习 MySQL 数据库。

这一节为读者介绍了 MySQL 数据库的知识。先从关系型数据库入手，介绍了什么是关系型数据库及关系型数据库所能实现的功能。然后，又通过列表的形式为读者介绍了 PHP 中常用的 MySQL 函数。熟练掌握这些函数对编写基于数据库的 Web 程序将起到事半功倍的效果。

这之后为读者介绍了 MySQL 数据库的相关操作包括如何连接到 MySQL 服务器，显示可用数据库，如何创建新的数据库，如何在库中建表及如何删除数据库等内容。数据操作的核心还是对表的操作。所以接下来就为读者详细介绍了如何对表执行 INSERT INTO 插入操作、SELECT 查询操作、UPDATE 更新操作、DELETE 删除操作以及 ALERT 修改表结构操作等关键内容。

其后又介绍了高级查询的使用，包括 WHERE 子句、LIMIT 子句、ORDER BY 子句的用法。最后，通过一个实际的例子，把本章所学的内容综合起来。对学习的内容做了一个很好的实际应用及总结。

通过本章的学习，相信读者对于如何使用 MySQL 与 PHP 结合创建 Web 应用程序都会有一个深刻的认识。

第15章

用 PHP 实现人机交互

- 用 Web 表单实现人机互动
- 表单元素基础知识
- 使用 URL 参数进行互动
- 表单使用实例

在本书的第 1 章曾就动态网页与静态网页的区别为读者做了介绍。其中动态网页的一个显著特征就在于它能实现人机互动。对用户输入或者选择的内容能做出相应的回应。这也是动态网页区别于静态网页的一个特征。其他动态技术是这样，PHP 当然也不例外。用 PHP 也能实现人机互动。这一节就为读者介绍如何用 PHP 实现人机互动。

人机互动一般采用两种方式：一种采用表单，通过表单不同的选项，或者输入不同的内容，返回结果也不同；另一种是采用 URL 地址加上各种参数实现互动，通过参数的不同，返回的内容也有所不同。基于这两种方式。本章内容主要也分为两大部分：用表单实现人机互动；URL 参数与 PHP。通过本章的学习，希望读者能够熟练地使用这两种方式实现人机互动。从而能在此基础上写出丰富的 Web 应用程序。

15.1　用表单实现人机互动

表单（FORM）是最常用的网页组件，同时也是交互式网页，实现人机互动的最常用的方式。不管是在动态的 ASP、PHP、JSP 文件中，还是在静态的 HTM、HTML 等文件中都可以发现表单的身影。这一节就来为读者介绍表单的使用以及如何使用表单来实现人机互动。

15.1.1　表单元素的组成

表单（FORM）由表单元素构成。常用的表单元素有以下几种：TEXT（文本框）、PASSWORD（密码输入框）、BUTTON（普通按钮）、RADIO（单选按钮）、CHECKBOX（复选框）、SELECT（列表框）、TEXTAREA（文本域）、SUBMIT（提交按钮）、RESET（重置按钮）、HIDDEN（隐藏域）等。下面分别对这些表单元素做简单的介绍。

1．TEXT（文本输入框）

文本框允许用户输入字符、数字或者中文字符等内容。它常用的属性有：NAME 属性、VALUE 属性、SIZE 属性等。

文本输入框通常代码如下：

```
<input type="text" name="username" value="在这里输入用户名" size="20">
```

其中，NAME 属性指该表单元素的名称，一是区别于其他表单元素的标志，另外设置该属性以便在程序中能够调用该表单元素。一个表单元素只有属于自己的 NAME（或者 ID）属性，才能被正常调用。VALUE 属性指代文本输入框的默认值，即在用户没有任何输入的情况下显示的值。SIZE 属性指该文本输入框在浏览器中表现出的长度，值越大，文本框就越长。在多数浏览器中，文本框的 SIZE 属性默认是 20 个字符。

2．PASSWORD（密码输入框）

基本属性与 TEXT（文本输入框）相同。不同的是文本输入框中，可以输入中文或者其他字符。而一旦 PASSWORD（密码输入框）获得光标，中文输入法将不再起作用。密码输入框里面只能输入键盘上所标出的符号，而不能输入中文。另外，TEXT（文本输入框）中所输入的内容全都是可见的，而 PASSWORD（密码输入框）中输入的所用信息都将以星号（*）显示。之所以这样就是为了保护密码不被别人发现。

3．BUTTON（普通按钮）

按钮表单像下面这样：

```
<input type="button" value="点击这里" onclick="functionname()">
```

该表单元素，也具有 NAME 属性与 VALUE 属性。属性的意义与 TEXT（文本输入框）大同小异。通常情况下 BUTTON（按钮）元素还会有 ONCLICK（单击）事件，该事件通常指向 JavaScript 函数。这样，在按钮按下的情况下就能够执行相应的操作。

4. RADIO（单选按钮）

单选按钮的使用如下所示：

```
<input type="radio" name="sex" value="男" checked>
```

单选按钮通常是以一组出现，该组应具有相同的 NAME 属性和不同的 VALUE 属性。在出现的一组单选按钮中，只允许用户选择一组中的一个，所以叫单选按钮。如果某个单选按钮的初始状态为选中状态，则该单选按钮有 checked 属性。

5. CHECKBOX（复选框）

复选框像下面这样的样式：

```
<input type="checkbox" name="fav" value="游戏">
```

该表单元素和 RADIO（单选按钮）类似，通常也是以一组出现，该组具有相同的 NAME 属性和不同的 VALUE 属性。不过与 RADION（单选按钮）不同的是，在一组中允许用户进行多项选择。所选择的结果都将被发送到指定的 URL 进行处理。

6. SELECT（列表框）

列表框通常使用以下所示的样式：

```
<select name="type" size="1">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
</select>
```

该表单元素与前面提到的几种有一定的区别。SELECT 元素有 NAME 属性也有 SIZE 属性。但是它的每个 VALUE 属性却包括在其下的每个<option>项里面，即每个<option>项对应一个 VALUE，不应有重复。另外根据 SIZE 属性的不同，SELECT（列表框）元素在浏览器中会有不同的形式。当 SIZE 等于 1 时，表现为下拉菜单的形式；当 SIZE 大于 1 时，则表现为列表框形式。

7. TEXTAREA（文本域）

文本域的使用情况像下面这样：

```
<textarea name="content" rows="5" cols="40">内容在这里输入</textarea>
```

TEXTAREA（文本域）又叫多行文本框，这是相对于 TEXT 而言。因为 TEXT 所输入的内容只能是一行，中间不允许也不会出现回车换行。而 TEXTAREA（文本域）则不然，里面可以随意使用回车键。表征该元素样式的属性有两个分别是 rows（行）与 cols（列）。rows（行）的大小决定了 TEXTAREA（文本域）的高度，而 cols（列）的大小则决定了 TEXTAREA（文本域）的宽度。另外，该表单元素也没有 VALUE 属性。如果要为文本域设定初始值，把想要设定的内容放在“<textarea>”标记与“</textarea>”标记之间即可。

8. SUBMIT（提交按钮）

提交按钮使用样式如下所示：

```
<input type="submit" value="提交">
```

提交按钮的作用是把表单输入结果发送到指定的 URL，当单击事件发生时，结果将会被提交。该表单元素也有 NAME 属性与 VALUE 属性。作用与上面介绍的元素的作用相类似，不再赘述。

9．RESET（重置按钮）

重置按钮也比较简单，像下面这样：

```
<input type="reset" value="重新填写">
```

RESET（重置按钮）。重置按钮的作用是把表单中所有填写好的内容恢复到原始值。当表单中的内容需要重新填写时单击该按钮。同时该元素也具有 NAME 与 VALUE 属性。

10．HIDDEN（隐藏域）

隐藏域的使用方法像下面这样：

```
<input type="hidden" name="hide" value="hideinfo">
```

HIDDEN（隐藏域）。如果有一些信息不方便或者不必要显示出来，使用 HIDDEN（隐藏域）是比较方便的。该元素并不会在浏览页面上显示。由于它不能显示的特性，所以通常情况下都要为 HIDDEN 指定一个 VALUE 值。并且该值是不能被用户改变的。而 NAME 属性则与上面介绍的其他表单元素类似。

15.1.2　在普通 Web 页中插入表单

在普通的 Web 页中插入表单的方法很简单。操作步骤如下所示。

（1）在网页的适当位置放置这样的代码：

```
<form name=f_name action=url method=get or post>
</form>
```

其中的 f_name 指该整个表单域的名称；URL 为处理该表单结果的 URL 地址。method 则表明采用何种方式发送表单域的数据。

（2）把所需要的表单元素放入到“<form>”标记与“</form>”标记之间即可。

例如下面的代码就在一个普通 Web 页中插入了一组表单。

```
<html>
<head>
<title>插入表单示例</title>
</head>
<body>
<center>
<h3>插入表单示例</h3>
<p>
<table border=1>
<form action=15-2.php method=post>
<tr>
```

```
<td>姓名：</td>
<td><input type=text name="name" size="12"></td>
</tr>
<tr>
<td>性别：</td>
<td>
<input type=radio name=sex value="男" checked>男
<input type=radio name=sex value="女">女
</td>
</tr>
<tr>
<td>生日：</td>
<td>
<select name=month>
<option value=1>1</option>
<option value=2>2</option>
<option value=3>3</option>
<option value=4>4</option>
<option value=5>5</option>
<option value=6>6</option>
<option value=7>7</option>
<option value=8>8</option>
<option value=9>9</option>
<option value=10>10</option>
<option value=11>11</option>
<option value=12>12</option>
</select>月
<select name=date>
<option value=1>1</option>
<option value=2>2</option>
<option value=3>3</option>
<option value=4>4</option>
<option value=5>5</option>
<option value=6>6</option>
<option value=7>7</option>
<option value=8>8</option>
<option value=9>9</option>
<option value=10>10</option>
<option value=11>11</option>
<option value=12>12</option>
<option value=13>13</option>
<option value=14>14</option>
<option value=15>15</option>
<option value=16>16</option>
<option value=17>17</option>
<option value=18>18</option>
<option value=19>19</option>
<option value=20>20</option>
<option value=21>21</option>
```

```
<option value=22>22</option>
<option value=23>23</option>
<option value=24>24</option>
<option value=25>25</option>
<option value=26>26</option>
<option value=27>27</option>
<option value=28>28</option>
<option value=29>29</option>
<option value=30>30</option>
<option value=31>31</option>
</select>月
</td>
</tr>
<tr>
<td>爱好：</td>
<td>
<input type=checkbox name=favior[] value="旅游">旅游
<input type=checkbox name=favior[] value="运动">运动
<input type=checkbox name=favior[] value="看电影">看电影<p>
<input type=checkbox name=favior[] value="游戏">游戏
<input type=checkbox name=favior[] value="上网">上网
<input type=checkbox name=favior[] value="听音乐">听音乐<p>
<input type=checkbox name=favior[] value="登山">登山
<input type=checkbox name=favior[] value="聊天">聊天
</td>
</tr>
<tr>
<td>其他：</td>
<td>
<textarea cols=24 rows=5 name=other>
其他信息在这里输入：
</textarea>
</td>
</tr>
<tr>
<td colspan="2">
<center>
<input type=submit value="提交">
<input type=reset value="重置">
</center>
</td>
</tr>
</form>
</table>
</center>
</body>
</html>
```

把以上代码保存为“15-1.php”（由于其中并没有出现任何 PHP 代码，完全是纯 HTML，所以也

可以保存为 HTML 文件）。在 PHP 执行环境下执行该 PHP 文件（如果保存的是普通 HTML 文件，也可以直接用浏览器打开），执行结果将会如图 15.1 所示。

以上代码中使用了多种表单元素。通过实例代码，及其执行结果，可以了解到，如何在普通 Web 页中插入表单元素。及如何使用各种表单元素。

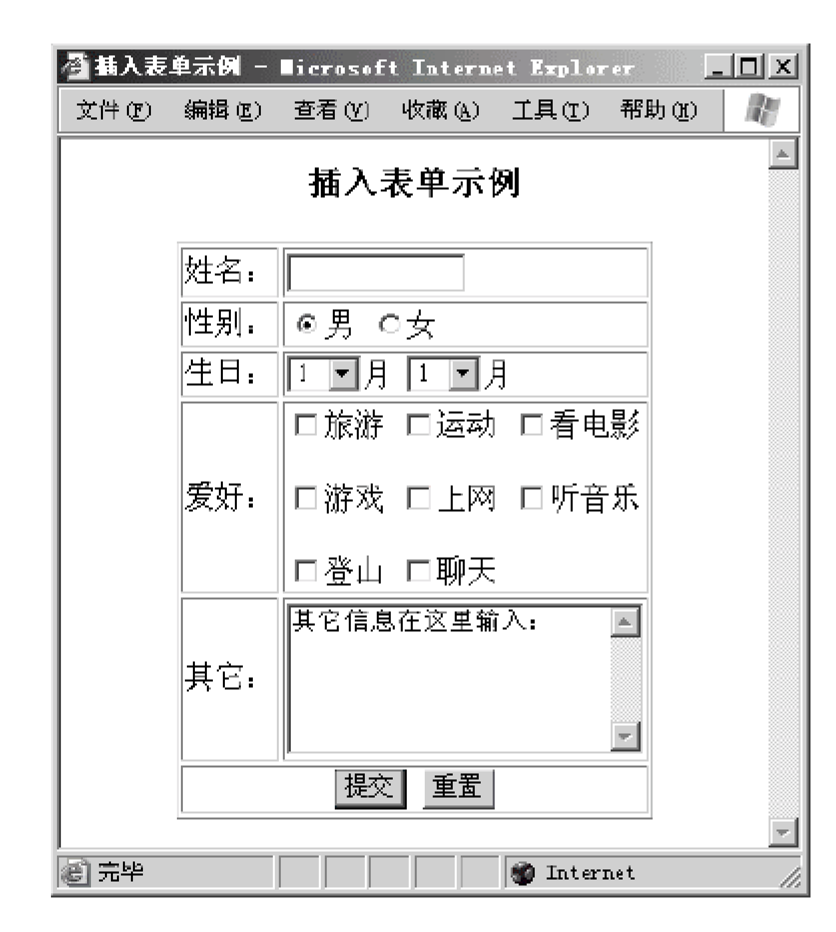


图 15.1　插入表单示例执行结果

15.1.3　更改表单的 ACTION 属性到 PHP 程序

只有了表单，相当于只有了前台，如果没有后台的支撑，只有表单的前台就什么也做不了。连接前台与后台之间的纽带就是表单（FORM）的 ACTION 属性。所以，要想让表单起作用，就必须有特定的后台处理程序；而要想让这后台处理程序为前台表单服务，就必须为表单指定 ACTION 属性。表单的 ACTION 属性通常都为一个 URL 地址。

在 15-1.php 中，form 标签没有任何属性，这里为其指定 ACTION=15-2.php（现在这个文件并不存在，在 15.1.5 小节将会创建）。修改以后备用。

15.1.4　表单 METHOD 属性 POST 与 GET 区别

在 15.1.2 小节就曾为读者介绍过，表单的 METHOD 属性指代表单数据的发送方式。通常表单数据通过两种方式发送到目标 URL。一种是 POST 方式，另一种是 GET 方式。这两种方式是有一定区别的。

理论上两者的区别：用 GET 方法提交的表单，提交的数据被附加到 URL 上，作为 URL 的一部分发送到服务器端，这种方式就类似于采用 URL 方式实现互动（后面将后讲到）；而 POST 方式则是将表单中的信息作为一个数据块发送到服务器端，这种方式不依赖于 URL。

形式上两者的区别：在实际应用时，两者的区别也是很明显的，由于 GET 方法提交的数据是被附加到 URL 上发送的，所以在地址栏中将会显示出用户所输入的数据作为参数附加在 URL 后面；反之由于 POST 方法不依赖于 URL，所以使用 POST 方法传送数据时，地址栏只会显示表单 ACTION 所指向的 URL，而不会显示用户输入的数据。

通过以上分析可以知道，两种方式的区别。在何种情况下使用何种数据提交方式呢？一般情况下，如果用户提交的信息不含私密性的东西，如使用者的用户名、密码等信息时，使用 GET 方式提交表单数据。反之，如果是论坛、聊天室或者其他 Web 程序的登录页，其中包含有使用者的用户名或者密码内容，这时就一定要使用 POST 来提交表单数据。

15.1.5　用 PHP 作后台处理表单提交数据

在普通 Web 页中插入了相应的表单元素，为 FORM 表单指定了 ACTION 属性，并且也使用了相应的 METHOD 提交方式。这一小节，就来创建对表单提交数据进行处理的后台程序。

表单提交后，一般的处理是对用户提交的数据进行分析，要存入文本文件，或者存入数据库，还是对用户提交的信息与已经存在的记录进行对比，以采取相应的动作。不过执行这些动作的前提是先要获得用户提交数据的内容。

这里要提到 php.ini 中的 GLOBAL 值。在老版本的 PHP 中，默认值是开启的。表现在 php.ini 中就是 GLOBAL=ON。这样就会存在一些问题，所有用户提交的变量，都可以当作全局变量直接被引用，如用户提交页中有如下一个表单。

```
<form method=post action=url>
请输入用户名：<input type=text name=username>
<p>
请输入密码：<input type=password name=pass>
<p>
<input type=submit value="确认提交">
</form>
```

如果此时 php.ini 中的 GLOBAL 值为 ON，则这些被提交的变量，usernma、pass 在目标页面中都可以这样被直接引用，$username、$pass。

如果是这样就会存在一定的安全隐患。所以在后来的 PHP 版本中，该 GLOBAL 项都被关闭了。如果该选项值为 OFF 的话，这些被提交的变量就不能被直接引用。要引用这些变量必须通过相应的 POST 或者 GET 数组。

如上面提交的表单在 php.ini 中的 GLOBAL 值为 OFF 时，就要这样引用：$_POST[username]、$_POST[pass]。如果 METHOD 属性为 GET，则要这样引用：$_GET[username]、$_GET[pass]。

以下讲到例子，及本书中所有的有关表单提交变量的情况均采用 GLOBAL=OFF 的值来设定。

准备工作做完，就开始来创建对 15-1.php 的处理页面即 15-2.php。这里并不对数据采取什么处理，只是先获取提交的内容并显示出来。

```
<?
if(!$_POST[name])                                //判断有无数据被提交
{
     echo "没有相应的用户名";
     echo "<p>";
     echo "点<a href=15-1.php>这里</a>返回";
     exit();                                     //中止所有 PHP 运行
}
else                                             //如果有数据提交，则显示相应 HTML
{
?>
<html>
<head>
<title>提交表单的处理示例</title>
</head>
<body>
<center>
<h3>提交表单处理示例</h3>
<table border=1>
<tr>
```

```
<td colspan=2>以下为用户提交内容</td>
<tr>
<td>姓名：</td>
<td><? echo $_POST[name]?></td>
</tr>
<tr>
<td>性别：</td>
<td><? echo $_POST[sex]?></td>
</tr>
<tr>
<td>生日：</td>
<td><? echo $_POST[month]."月".$_POST[date]."日"?></td>
</tr>
<tr>
<td>爱好：</td>
<td>
<?
for($i=0;$i<count($_POST[favior]);$i++)
{
	echo $_POST[favior][$i];
	echo "、";
}
?>
</td>
</tr>
<tr>
<td>其他：</td>
<td>
<?
$other=htmlspecialchars($_POST['other']);		//去掉特殊字符
$other=ereg_replace("\r\n","<br>",$other);		//转行回车
$other=ereg_replace("\r","<br>",$other);		//转行换行
echo $other;
?>
</td>
</tr>
<tr>
<td colspan=2>
<center><a href=15-1.php>返回</a></center>
</td>
</tr>
</table>
</center>
</body>
</html>
<?
}
?>
```

把以上代码保存为“15-2.php”。把该文件与 15-1.php 放入同一个目录之下。然后，先在 PHP 执行环境下运行 15-1.php（记得在运行前，要把 15-1.php 中 FORM 的 METHOD 属性设为 POST，ACTION 属性设为 15-2.php）。按要求完整填写相应信息，然后单击“确认提交”按钮。执行结果将会如图 15.2 所示。

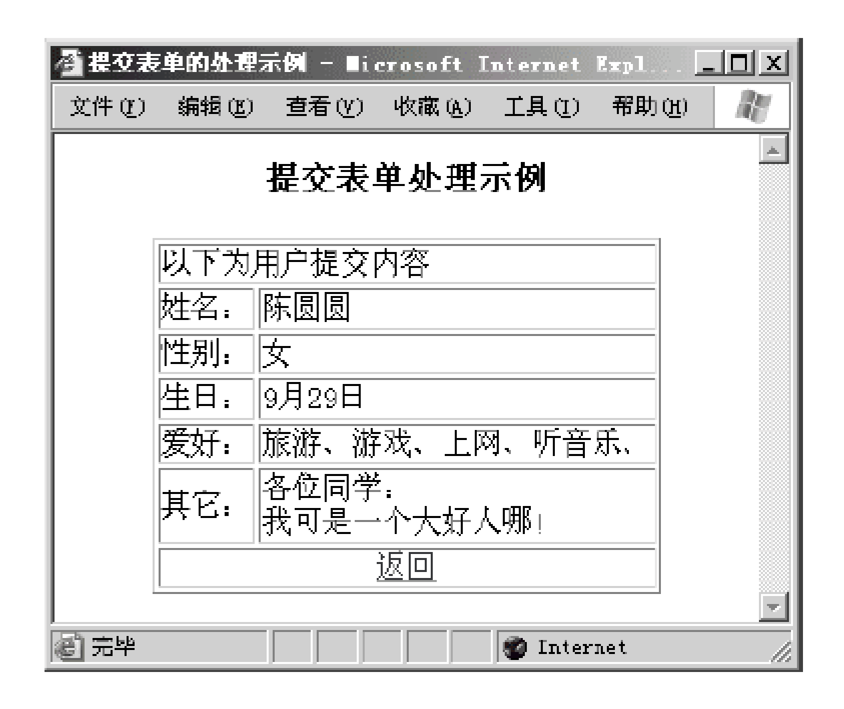


图 15.2　提交表单处理的示例执行结果

从图 15.2 可以发现，正确的返回了用户提交的各项信息。这是对这些信息进行下一步操作的关键。

需要注意的就是多选框 checkbox 的值的获取。在命名表单时使用 name[]这样的形式。这样在提交时，提交的内容就会以 name[0]、name[1]、……、name[n]的形式出现。获取时也要使用循环来遍历数组。

对于如何获取提交表单数据，这一节就介绍到这里。

15.2　URL 参数与 PHP

使用 URL 参数，也是与用户实现互动的一种重要方式。不过该种方式与使用表单相比缺点也显而易于，就是可视化及友好程度都要差一些。表单使用各种表单元素来供用户进行输入信息或者选择某些选项。而 URL 则只能提供超连接的形式来供用户选择或者使用某些信息。同时由于 URL 的数据发送方式与表单提交的 GET 方式类似，所以使用 URL 所附带的各种参数都会完全显示在浏览器的地址栏上。所以，它在安全性方面也存在局限性。

15.2.1　在 PHP 的 URL 地址上加入参数

使用 URL 参数方式传递数据，就是在 URL 地址后面加上适当的参数。URL 实体将对这些参数进行处理。简单的使用方法这像下面这样：

```
http://127.0.0.1/15-3.php?username=JACK&sex=男&age=23
```

然后，直接在地址栏中输入这些内容即可（由于还没有编写相应的 PHP 文件，所以现在执行还不会有任何如果）。

通过以上代码了解到，在 URL 地址后面加参数，要在完整的 URL 后加“?”符号。然后是参数名称=参数值，如上面的 username=JACK。多个参数之间要使用“&”符号，它表示参数与参数之间的连接。

15.2.2　用 PHP 处理提交的参数

前面在介绍表单的提交方式时曾提到过：采用表单的 GET 提交方式提交的数据被附加到 URL 上，作为 URL 的一部分发送到服务器端。这种方式就类似于采用 URL 方式实现互动。所以对于采用 URL

参数方式提交表单数据的获取也采用$_GET["参数名称"]的方式来获取。

如上面的：

http://127.0.0.1/15-3.php?username=JACK&sex=男&age=23

调用时就可以用$_GET[usernma]、$_GET[sex]、$_GET[age]来获取。

下面来实际编写一个可以处理用户以 URL 加参数形式提交内容的 PHP 页面。具体内容参考以下代码：

```
<?
if(!$_GET[username] )                               //判断有无数据被提交
{
    echo "没有相应的用户名";
    echo "<p>";
    exit();                                         //中止所有 PHP 运行
}
else                                                //如果有数据提交，则显示相应 HTML
{
?>
<html>
<head>
<title>提交 URL 参数的处理示例</title>
</head>
<body>
<center>
<h3>提交 URL 参数处理示例</h3>
<table border=1>
<tr>
<td colspan=2>以下为用户提交内容</td>
<tr>
<td>姓名：</td>
<td><? echo $_GET[username]?></td>
</tr>
<tr>
<td>性别：</td>
<td>
<?
if($_GET[sex])
echo $_GET[sex];
else
echo "没有相应的性别";
?>
</td>
</tr>
<tr>
<td>年龄：</td>
<td><?
if($_GET[age])
echo $_GET[age];
```

```
else
echo "没有相应的年龄";
?>
</td>
</tr>
</table>
</center>
</body>
</html>
<?
}
?>
```

把以上代码保存为“15-3.php”。然后在 PHP 执行环境中运行以上代码，将输出如图 15.3 所示的执行结果。

为什么会这样？因为上面的代码使用了有无参数提交的判断。而上面在执行时并没有附加任何参数，所以才会出现如图 15.3 所示的执行结果。下面带上参数再运行一次，在地址栏中输入下面的内容。

```
http://127.0.0.1/15-3.php?username=JACK&sex=男&age=23
```

然后的执行结果就会如图 15.4 所示。



图 15.3　提交 URL 参数的处理示例（无参数）

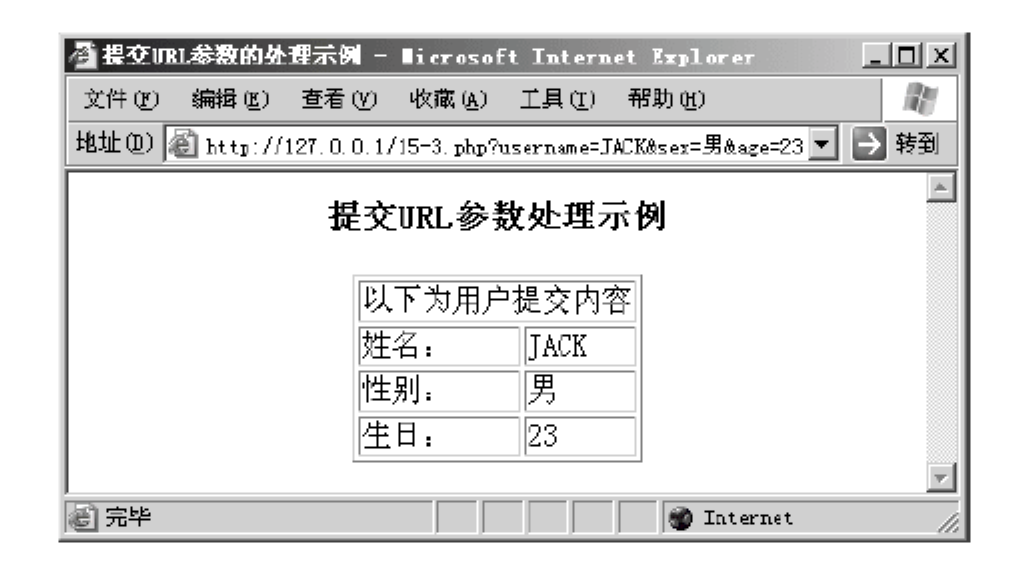


图 15.4　提交 URL 参数的处理示例（有参数）

通过图 15.4 可以发现，在加上相应的参数之后，程序正确显示出了用户所提交的内容，实现了互动的目的。

15.3　表单使用实例

这一节通过一个实例来向读者介绍如何使用表单。与前面几小节介绍的不同，这一小节的实例将会把表单输入的前台与表单数据处理的后台整合在一个文件之中。把表单相关的前台与后台整合起来，一方面可以减少文件个数，另一方面也便于管理，而且当文件需要移动时，只用移动一个文件即可。

该实例实现的如下功能，当有用户表单提交操作时，把用户所输入的表单内容存放到一个以该用户所输入的用户名为文件名的一个文本文件。即该文本文件，存储着该用户的相关信息。

具体内容参看以下代码：

```
<?
if(!$_POST[name])                              //判断有无数据被提交
{                                              //如果没有数据被提交，则显示提交表单内容
?>
<html>
<head>
<title>表单使用实例</title>
</head>
<body>
<center>
<h3>表单使用实例</h3>
<p>
<table border=1>
<form method=post action=<?echo $PATH_INFO?>>
<tr>
<td>姓名：</td>
<td><input type=text name="name" size="12"></td>
</tr>
<tr>
<td>性别：</td>
<td>
<input type=radio name=sex value="男" checked>男
<input type=radio name=sex value="女">女
</td>
</tr>
<tr>
<td>生日：</td>
<td>
<select name=month>
<option value=1>1</option>
<option value=2>2</option>
<option value=3>3</option>
<option value=4>4</option>
<option value=5>5</option>
<option value=6>6</option>
<option value=7>7</option>
<option value=8>8</option>
<option value=9>9</option>
<option value=10>10</option>
<option value=11>11</option>
<option value=12>12</option>
</select>月
<select name=date>
<option value=1>1</option>
<option value=2>2</option>
<option value=3>3</option>
<option value=4>4</option>
<option value=5>5</option>
<option value=6>6</option>
```

```
<option value=7>7</option>
<option value=8>8</option>
<option value=9>9</option>
<option value=10>10</option>
<option value=11>11</option>
<option value=12>12</option>
<option value=13>13</option>
<option value=14>14</option>
<option value=15>15</option>
<option value=16>16</option>
<option value=17>17</option>
<option value=18>18</option>
<option value=19>19</option>
<option value=20>20</option>
<option value=21>21</option>
<option value=22>22</option>
<option value=23>23</option>
<option value=24>24</option>
<option value=25>25</option>
<option value=26>26</option>
<option value=27>27</option>
<option value=28>28</option>
<option value=29>29</option>
<option value=30>30</option>
<option value=31>31</option>
</select>月
</td>
</tr>
<tr>
<td>爱好：</td>
<td>
<input type=checkbox name=favior[] value="旅游">旅游
<input type=checkbox name=favior[] value="运动">运动
<input type=checkbox name=favior[] value="看电影">看电影<p>
<input type=checkbox name=favior[] value="游戏">游戏
<input type=checkbox name=favior[] value="上网">上网
<input type=checkbox name=favior[] value="听音乐">听音乐<p>
<input type=checkbox name=favior[] value="登山">登山
<input type=checkbox name=favior[] value="聊天">聊天
</td>
</tr>
<tr>
<td>其他：</td>
<td>
<textarea cols=24 rows=5 name=other>
</textarea>
</td>
</tr>
<tr>
```

```
<td colspan="2">
<center>
<input type=submit value="提交">
<input type=reset value="重置">
</center>
</td>
</tr>
</form>
</table>
</center>
</body>
</html>
<?
}
else                                                    //如果有数据提交，则执行相关操作
{
    $name=$_POST[name];
    $sex=$_POST[sex];
    $birthday=$_POST[month]."月".$_POST[date]."日";
    for($i=0;$i<count($_POST[favior]);$i++)
    {
        $f=$f.$_POST[favior][$i]."+";
    }
    $other=htmlspecialchars($_POST['other']);           //去掉特殊字符
    $other=ereg_replace("\r\n","<br>",$other);          //转行回车
    $other=ereg_replace("\r","<br>",$other);            //转行换行
    $s=$name."``".$sex."``".$birthday."``".$f."``".$other."\n";
    $myfile=fopen($name,"w");                           //创建文件
    fwrite($myfile,$s);                                 //置入相应内容
    fclose($myfile);                                    //关闭文件
    echo "用户输入内容为：";
    echo $name."<p>";
    echo $sex."<p>";
    echo $birthday."<p>";
    echo $f."<p>";
    echo $other;
    echo "<p>";
    echo "用户文件已经创建完毕！";
}
?>
```

把以上代码保存为“15-4.php”，然后在 PHP 执行环境中运行该 PHP 文件。将会出现与 15-1.php 基本类似的执行结果。因为它们采用的前台内容及格式基本上是相同的，区别的是在后台。

在执行界面上的表单中，输入相应内容。这里输入以下内容（当然读者也可以根据自己喜好，输入相应内容）：

姓名：JACK、性别：男、生日：11 月 5 日、爱好：运动+看电影+听音乐+登山、其他：该同志为美籍华人。\n现住美国纽约市。

输入完毕后，单击“确认提交”按钮，将会出现如图 15.5 所示的执行结果。

从图 15.5 的执行结果可以发现，程序正确显示出了用户输入的各项内容，并且还提示用户文件已经创建完毕。

然后到 15-4.php 所在目录里能够发现名为 JACK 的文件，用记事本打开它。里面有以下内容：

**JACK "男" 11 月 5 日 "运动+看电影+听音乐+登山+" 该同志为美籍华人。
现住美国纽约市。**

说明用户文件已经被正确创建。

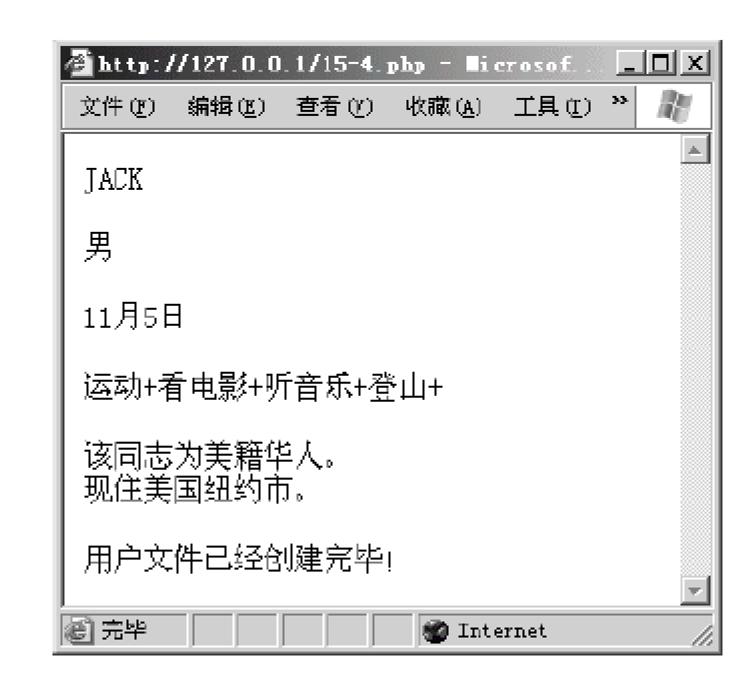


图 15.5　表单使用实例处理结果

这个表单使用的实例就为读者介绍到这里。具体使用中 Web 表单还有更多的用途，如把用户记录存入数据库、获得用户提交信息与库中内容相比较、通过用户提交表单执行相关操作等。可以说表单的使用非常广泛。所以只有熟练掌握了表单的使用，才能开发出高效的人机交互 Web 程序。

15.4　本 章 小 结

实现人机交互是动态网页技术的重要特点之一。本章主要介绍用 PHP 实现人机交互的两种方式，表单方式和 URL 参数方式。其中表单方式常用于论坛中的用户注册、用户登录信息录入等。而 URL 参数方式更多的见于留言本中的分页显示、论坛中的分主题显示、论坛中不同子论坛之间的跳转等。可以说，两种方式都有着广泛的应用，并且它们各有其优缺点，互为补充。

通过本章对两种实现人机交互方式的学习，及对具体使用实例的介绍，相信读者对于如何通过 PHP 代码实现人机交互，会有一个深刻的认识。从而为写出高效率的 Web 应用程序打下坚实的基础。

编写人机交互程序有一个问题需要引起读者的注意，那就是要了解 php.ini 中 GLOBAL 的状态，对于该值不同的状态采用不同的变量获取形式。

第3篇



PHP 常用模块

第16章　计数器程序

第17章　网上投票程序

第18章　文本留言板程序

第19章　PHP博客程序

第16章 计数器程序

- ⏭ 计数器的原理
- ⏭ 图形化计数器的实现
- ⏭ 使用 Cookie 实现防刷新
- ⏭ 多用户计数器的实现

计数器在网络上的运用非常广泛，比较常见的应用的是访客浏览量计数器。在自己的网站里放置一个访客浏览量计数器，可以帮助站长了解站点的访问情况。本章就来介绍怎样使用 PHP 来开发计数器小程序。其中包括简单计数器、计数器原理、计数器算法的设计、具体代码实现计数器、图形化计数器、设计算法、用图片替换文本、实际代码实现图形化计数器、计数器的进一步完善和实现多用户计数器等内容。

16.1　简单计数器

计数器的功能虽然单一，但根据存储及表现方式的不同也有多种多样。有简单的文本计数器、在文本计数器基础上的图形计数器、以数据库为基础的数据库计数器等。本节先来介绍简单的文本文件计数器。

16.1.1　计数器的原理

计数器的原理很简单，就是累加。如果原始值是“0”，就在 0 在基础上累加 1，结果就是“1”。每浏览一次就在原来浏览量的基础上累加 1。

既然是简单的文本计数器，就采用文本文件作为载体。第 1 次运行程序时，先判断有没有相应的文本文件，如果没有就创建文件，并置入“0”。以后程序运行时，先读出文本文件中储存的数值，把原值自增 1 再把增加的值显示出来。然后把增加的值重新写入文本文件即可。

16.1.2　设计算法

下面设计计数器算法。

把文本文件名定义给变量。先判断文件是否存在，这里要用到 fileexists()函数。如果不存在使用读写方式创建文件，这里要使用 open("文件名","a")函数。置入“0”，这里要使用 fwrite()函数。然后使用 file("文件名")函数读出文件数据内容，给原值加 1。使用 echo 函数显示到用户界面，然后再把增加过的值重新写入文本文件，关闭文件结束程序。

16.1.3　代码实现

下面具体来实现 16.1.2 节中所要求的功能。具体内容参看以下代码：

```
<html>
<head>
<title>PHP 文本计数器</title>
</head>
<body>
<?
$c_file="counter.txt";                              //文件名赋值给变量
if(!file_exists($c_file))                           //如果文件不存在的操作
{
      $myfile=fopen($c_file,"w");                   //创建文件
      fwrite($myfile,"0");                          //置入“0”
      fclose($myfile);                              //关闭文件
}
$t_num=file($c_file);                               //把文件内容读入变量
```

```
$t_num[0]++;                                  //文件内容自增 1
echo "欢迎！您是本站第".$t_num[0]."位访客！";   //显示文件内容
$myfile=fopen($c_file,"w");                   //打开文件
fwrite($myfile,$t_num[0]);                    //写入新内容
fclose($myfile);                              //关闭文件
?>
</body>
</html>
```

保存以上代码为“16-1.php”，然后在 PHP 运行环境中执行以上 PHP 文件，第 1 次运行的执行结果将如图 16.1 所示。

同时打开程序运行的文件夹，会发现多了一个叫“counter.txt”的文件。说明程序正常运行。刷新该页面，相当于程序重新加载一次，执行结果如图 16.2 所示。



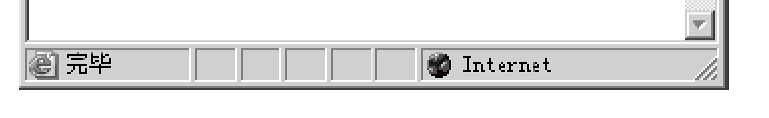


图 16.1　文本文件计数器第 1 次运行结果

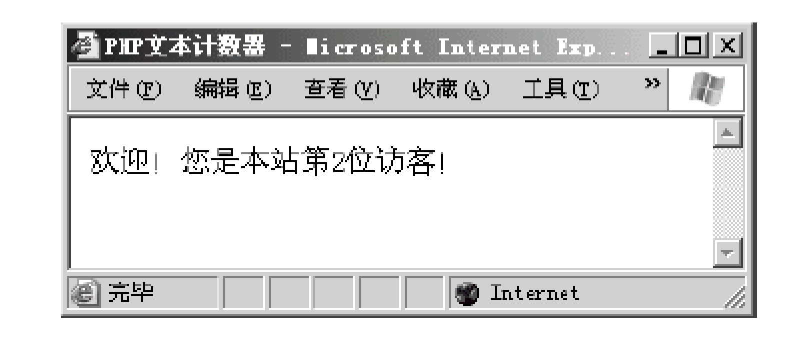


图 16.2　文本文件计数器第 2 次运行结果

说明程序已经正常运行，并已经能实现累加，经得起多次考验。

这一节简单文本计数器就介绍到这里。16.2 节将介绍图形化计数器。

16.2　图形化计数器

文本计数器虽然简单易行，但是美观上就要欠缺一点，如果采用图形的方式来表达计数器内容，就既能实现计数功能，又让人印象深刻。本节就来讲解在 PHP 中怎样在文本计数器的基础上实现“图形化”的计数器。

16.2.1　设计算法

既然图形计数器是在文本计数器的基础上建立起来的。所以它们的算法是完全相同的，都是简单的累加。不同的是文本计数器只是把文本内容简单的表现给用户，而图形计数器则是采用图片的方式来表达文本的内容而已。

16.2.2　用图片替代文本

在讲解本小节实现图形化计数器之前，请先准备内容为 0~9 的 10 张小图片，文件名要与内容一致分别为 0.gif、1.gif、……、8.gif、9.gif。

要实现图片替代文本，一位数是不会出现任何问题。但是超过一位就有麻烦了，不能准备 11~99 的图片，所以要把文本逐个分开，再分别用相应的图片来替换。这样的话，file()函数就不能满足需要

了，因为它是把文件按行读入数组的，所以这里要采用 getc()这个函数，因为这个函数是按字符来读取文件内容的。

16.2.3　代码实现

这一小节，来具体讲解一下怎么用代码来实现图形化计数器，同时也来对比一下 file()与 getc()两个函数的区别。

```
<html>
<head>
<title>PHP 图形化计数器</title>
</head>
<body>
<?
$c_file="counter.txt";                              //文件名赋值给变量
if(!file_exists($c_file))                           //如果文件不存在的操作
{
    $myfile=fopen($c_file,"w");                     //创建文件
    fwrite($myfile,"0");                            //置入“0”
    fclose($myfile);                                //关闭文件
}
$t_num=file($c_file);                               //把文件内容读入变量
$t_num[0]++;                                        //原始数据自增 1
$myfile=fopen($c_file,"w");                         //写入方式打开文件
fwrite($myfile,$t_num[0]);                          //写入新数值
fclose($myfile);                                    //关闭文件
echo "欢迎！您是本站第";                              //显示内容头部
$myfile=fopen($c_file,"r");                         //以只读方式打开文件
while(!feof($myfile))                               //循环读出文件内容
{
    $num=fgetc($myfile);                            //当前指针处字符赋值给变量
    if($num)                                        //如果数值存在执行操作
    {
        echo "<img src=images\\".$num.".gif>";      //显示相应图片
    }
}
fclose($myfile);                                    //关闭文件
echo "位访客！";                                     //显示内容尾部
?>
</body>
</html>
```

保存以上代码为“16-2.php”，在 PHP 运行环境下执行这个 PHP 文件，执行结果如图 16.3 所示。

说明该计数器程序在 1 位数的情况下还是能够正常运行，也是能经得起考验的。下面持续刷新，直到该计数器达到 9 以上，执行结果如图 16.4 所示。

执行结果表明，即使是在游览量为两位的情况下，该计数器程序依然是经得起考验的。

通过这一小节的介绍，一个既简单，又美观的图形化计数器已经完成了。还可以去掉代码中的 HTML 部分，只保留 PHP 代码，这样，就可以在其他 PHP 页面中引用这个文件了。



图 16.3　图形化计数器执行结果



图 16.4　图形化计数器在浏览量超过两位时的执行结果

16.3　添加“防止恶意刷新”功能

虽然现在已经实现了计数器的功能，但是有一点不太完美的是每当用户刷新一下就会使计数器运行一次。这样获得的情况往往不太准确，因为可能有别有用心的人不停地刷新，以达到计数器疯狂增加的目的。怎么样避免这种状况的发生呢？这里就要用到功能强大的 Cookie 了。这一小节来讲解怎样通过 Cookie 来完善图形化计数器。

16.3.1　设计算法

防刷新功能的算法是这样，先判断有没有 Cookie，如果没用 Cookie 就启动一次计数器并且写入当天系统日期到 Cookie 数据。当用户刷新或者第二次浏览时，就判断 Cookie 的日期是否与系统日期一致，如果两者一致则只读出原始数据，并不执行自增这一步。这样不论用户如何刷新，计数器就只能够显示，而不能增加了。

16.3.2　代码实现

这一小节，将通过具体的代码来实现增强版的计数器，通过此例也能加深读者对 Cookie 的认识。由于 setcookie 在使用之前不能有任何的输出操作，所以本小节的代码要对上节的图形化计数器代码做较大改动。

```
<?
$c_file="counter.txt";                          //文件名赋值给变量
if(!file_exists($c_file))                       //如果文件不存在的操作
{
    $myfile=fopen($c_file,"w");                 //创建文件
    fwrite($myfile,"0");                        //置入“0”
    fclose($myfile);                            //关闭文件
}
$t_num=file($c_file);                           //把文件内容读入变量
if($_COOKIE["date"]!="date(Y 年 m 月 d 日)")    //判断 COOKIE 内容与当前日期是否一致
{
$t_num[0]++;                                    //原始数据自增 1
```

```
$myfile=fopen($c_file,"w");                                    //写入方式打开文件
fwrite($myfile,$t_num[0]);                                     //写入新数值
fclose($myfile);                                               //关闭文件
setcookie("date","date(Y 年 m 月 d 日)",time()+60*60*24); //重新将当前日期写入 COOKIE 并设定 Cookie 的有效
期为 24 小时
}
?>
<html>
<head>
<title>PHP 图形化计数器</title>
</head>
<body>
<?
echo "欢迎！您是本站第";                                      //显示内容头部
$myfile=fopen($c_file,"r");                                    //以只读方式打开文件
while(!feof($myfile))                                          //循环读出文件内容
{
    $num=fgetc($myfile);                                       //当前指针处字符赋值给变量
    if($num)                                                   //如果数值存在执行操作
    {
        echo "<img src=images\\".$num.".gif>";                 //显示相应图片
    }
}
fclose($myfile);                                               //关闭文件
echo "位访客！";                                               //显示内容尾部
?>
</body>
</html>
```

保存以上代码为“16-3.php”，在 PHP 运行环境中执行以上代码。会发现虽然第 1 次能正确显示，但当刷新页面的时候，显示的还是原来的值。因为第 2 次执行时 Cookie 中的值与当前日期一致，程序并不执行自增操作，所以显示的还是原始值。自此该增强型的防恶意刷新计数器完成。

16.4　多用户计数器

在网上经常能找到面向多用户的计数器。任何人都可以注册一个用户名，然后系统给出一段 SCRIPT 代码，只要把代码贴到自己的页面上就可以实现计数器功能。那么怎样才能实现多用户的计数器呢？这一节就来介绍如何在 PHP 环境中编写多用户计数器。

16.4.1　多用户的原理

要想实现多用户计数器，就要有多个记录，并且各个记录之间要互不影响。访问其中一个用户的记录，只在此用户的原有记录上自增，然后重新保存在该用户记录里即可。这样就可以实现多用户计数器了。

16.4.2　实现方法

通常有两种方法可以实现多用户计数器。实际上也就是两种存储机制的问题。大致可以有如下两种方法：

方法 1。像前几节介绍的计数器一样，采用文本文件作为存储载体。每个用户记录对应一个文本文件。第 1 次访问时先判断该用户的文件是否存在，如果不存在就创建文件并写入“0”。如果文件存在就读出原值并自增“1”。显示新值，把自增后的值写入用户文件即可。

方法 2。采用数据库作为存储载体。每个用户对应表里的一条记录。记录内容为相应的用户及访问量。第 1 次访问时先判断表中有无该用户的记录，如果没有就写入记录并初始化访问量为“0”。如果记录存在就读出记录中访问量的值并自增“1”。显示出新值，再把自增后的值写入表中用户记录。

关于方法 1，与前面介绍的例子差不多，只是多了一个文件名的判断，这里不再多介绍，有兴趣的读者可以自己研究。这里重点介绍方法 2 的实现。

16.4.3　代码实现

这一节，将通过具体的代码来介绍数据库版的面向多用户计数器。通过此例也能加深读者对数据库的操作。由于要用到数据库、表，所以要先在库里创建表 counter。表应包含有字段 id、username、count 等内容。

第 1 步，运行以下代码创建 mysql 表：

```
<?
$myconn=mysql_connect("localhost","root","");                    //连接到服务器
mysql_select_db(test,$myconn);                                   //连接到 test 库
$query="create table counter (id int(5) not null auto_increment primary key,username varchar(20) not null,count
int(5) not null)";                                               //创建 counter 表语句
mysql_query($query);                                             //执行语句
mysql_close($myconn);                                            //关闭对数据库的连接
echo "你已经成功创建数据表";                                      //创建成功提示
?>
```

保存以上代码为“16-4.php”。先打开 MySQL 服务，然后在 PHP 运行环境下执行以上代码。如果成功就会提示创建数据表成功提示。

第 2 步，以下代码为多用户计数器的核心文件。

```
document.write("<?
if($_GET["username"])
{
$username=$_GET["username"];
$myconn=mysql_connect("localhost","root","");                          //连接到服务器
mysql_select_db(test,$myconn);                                         //连接到 test 库
$sqlstr="select * from counter where username='$username'";            //查询用户名语句
$result=mysql_query($sqlstr) or die(mysql_error());                    //执行查询语句
```

```
$num=mysql_num_rows($result);                                            //查询结果保存到变量
if($num==0)                                                              //如果结果为 0 执行操作
{
    $sqlstr="insert into counter (username) values ('$username')";       //插入记录语句
    mysql_query($sqlstr) or die(mysql_error());                          //执行语句
}
$sqlstr="select count from counter where username='$username'";          //重新查询
$result=mysql_query($sqlstr) or die(mysql_error());                      //执行查询语句
$count=mysql_fetch_array($result);                                       //结果保存到变量
$count[0]++;                                                             //自增 1
echo $count[0];                                                          //显示新结果
$sqlstr="update counter set count=count+1 where username='$username'";   //更新数据表
$result=mysql_query($sqlstr) or die(mysql_error());                      //执行更新语句
mysql_close($myconn);                                                    //关闭数据库连接
}
else echo "用户名不能为空";
?>
位访问者");
```

保存以上代码为“16-5.php”。然后在需要调用的 Web 页中插入以下代码：

```
<script language="javascript" src="http://127.0.0.1/16-5.php?username=test">
</script>
```

其中的 src 放置 counter.php 文件的绝对路径，username 为用户名。然后再浏览用户的普通 Web 页面就可以顺利执行，运行效果如图 16.5 所示。

这里可以更改 username 以验证是否支持多用户。更改用户名为 test2，可以发现运行效果与用户名为 test 类似。只不过又从 1 开始计数。



图 16.5　多用户计数器测试结果



图 16.6　多用户计数器在更改用户后的执行结果

用 PHPMYADMIN 打开数据库，查看 TEST 库中的 counter 表，发现已经有了两条记录，一条为 test，另一条为 test2。说明多用户计数器已经完成了。

16.5　本 章 小 结

本章为读者介绍了能经常用到的 PHP 计数器。详细说明了在 PHP 编程环境下，怎样开发计数器。通过本章学习，读者不仅学到了计数器的算法，同时也复习了 PHP 对文件的操作、PHP 中 Cookie 的使用、PHP 对 MySQL 数据库的操作等内容。计数器的功能虽然简单，但要做一个完善的计数器，就要结合多方面知识才能实现的。

第17章

网上投票程序

⏭ 投票程序的原理

⏭ 投票项管理

⏭ 显示投票项

⏭ 防止重复投票

投票程序在网上也有着广泛的应用。如对某一个问题的问卷调查，或者网站站长对网站满意度的调查等。本章将向读者介绍如何使用 PHP 来制作一个投票系统。通常的投票系统按数据存储机制的不同可分为文本类和数据库类，本章将重点介绍后者。本章内容包括网上投票程序的原理、数据库表的设计、管理页面的编写、用户提交内容的处理等。通过对本章的学习，读者将能够轻松地使用 PHP 编写出投票程序。

17.1　投票程序的原理

投票程序的实现方法其实并不复杂，通常采用这样的算法：事先建立好投票项，当用户通过前台对某一项（适用于单项选择）或者几项（适用于多项选择）投票项进行选择并提交时，后台处理就在相应的记录上增加 1。显示投票内容时一并显示相应投票项的记录情况。这样就可以实现一个投票程序了。

当然单选项的计票方式就很简单，每投一票计数一次即可。不过这里涉及一个问题，那就是关于多选项如何计票的问题。可以把一次多选的所有选择项当作一次计数，如一次多选了三项，就当作一次计数。也可以把一次多选的几项当作几次计数，如一次多选选了三项内容，就当作三次计数。本章的实例将采用后一种方式来计票。

17.2　本实例的特点

本章将向读者介绍一个简单的网上投票系统。本系统大致有如下几个特点。

- 用户可以无限增加投票项目，每个项目可以设定为单选或者多选。
- 每个项目也可以自定义投票项数目、投票项内容。
- 投票结果采用图片数字百分比两种方式显示，用户也可以自定义每个投票项的图片颜色。
- 投票显示采用 JS 方式调用，所以，可以在普通 Web 页里插入相应的 JS 代码来显示。
- 可以设置是否用 Cookie 保存投票人的 IP 记录，同时设置是否允许同一 IP 地址的多次投票，这样可以防止重复投票。

以上就是本章所要介绍的投票系统的主要特点，17.3 节将为读者介绍数据表的结构及各项内容。

17.3　投票实例数据表的设计

数据表的设计是一个数据库应用程序的最为重要的一个环节。因为数据表设计的适当与否直接关系着程序的运行效率。良好的、设计合理的数据表将对程序的运行起到事半功倍的效果。相反，如果数据表设计得混乱那么程序运行起来，要完成同一个操作就需要执行更多的指令。这样程序运行就会受到相当大的影响。

为了简化操作，使程序更容易理解，本章的投票程序实例的数据表采用两个表来设计。采用两个表来包含整个系统的全部信息势必比采用一个表设计要提高程序的复杂程度，但这样会使得思路清晰，使读者特别是初学数据库技术的读者容易理解。

本章的投票程序实例共有两个表，详见以下内容。

投票项记录表，表名 VOTE，共有 4 个字段，每个字段及其含义见表 17.1。

表 17.1　投票项记录表 VOTE 字段列表

字　段　名	类　　别	长　　度	内　　容	其　　他
ID	数值型	5	记录投票项的ID号	表的主关键字，有自动增加的特性
V_NAME	字符型	20	投票项的名称，实际上是一个投票项的标题	无
V_NUM	数值型	5	记录投票项的结果。该字段存放投票的总数。如果投票项为单选项，则每有一次投票，该值增加1。如果投票项为多选项，则每有一次投票，该值增加投票选择项的项目数	无
V_TYPE	数值型	1	记录投票项的类型。该字段保存投票类型的标记。如果该值为0，表示为单选项，如果该值为1，表示为多选项。在显示投票时将调用该值以确定显示的表单元素是radio还是checkbox。然后就在处理投票结果时调用该值，如果值是单选项，则采用一次计数；如果是多选项则采用多次计数	无

以上是记录投票项的表，下面是记录每一个投票项的选择支表。

投票项选择支表，表名 RECORD，共有 5 个字段，每个字段及含义见表 17.2。

表 17.2　投票项选择支表 RECORD 字段列表

字　段　名	类　　别	长　　度	内　　容	其　　他
ID	数值型	5	记录投票项的ID号，以此对每个投票选择支进行区别	表的主关键字，有自动增加的特性
R_ID	数值型	5	记录该投票支所对应的VOTE中投票项的ID，以表明该选择支是属于哪一个投票项的	无
R_NAME	字符型	50	记录选择支的内容，实际上存放一个投票选择项的内容	无
R_NUM	数值型	5	记录该选择支的被选结果，存放该选择支被选择的次数	无
R_COLOR	数值型	1	记录该选择支的显示颜色，实际是用相应的小图片来显示	无

由于采用的是两个表设计，所以，VOTE 表中存放投票项的总情况，具体每一项内容存放到 RECORD 表中。这样的表结构虽然比采用单一表设计的表稍显复杂，但实际运行起来，却会比采用单表的设计效率更高。

17.4　代 码 实 现

本节就开始通过具体的代码来一步一步完成这个投票程序。该投票程序共有以下几个模块组成：

新建投票项、显示所有投票项、单一投票项的 JS 显示、处理用提交的内容等。下面，就分别介绍。

17.4.1　准备工作

在做这些工作之前，有两项工作是必须要事先完成的。

第 1 步，建立一个配置文件，存放程序所要调用的一些重要变量，如主机名、用户名、用户密码、库名、表名等内容。其内容如下：

```
<?
$db_host="localhost";                    //主机名
$db_user="root";                         //用户名
$db_pass="";                             //密码
$db_name="test";                         //操作库名
$table_vote="vote";                      //表名 1
$table_record="record";                  //表名 2
$re_vote=false;                          //是否允许重复投票
?>
```

把以上代码保存为“17-1.php”，该文件将作为整个投票程序的配置文件在多处被调用，所以要首先被创建。

第 2 步，由于所有的操作都需要通过对相应的表的操作来完成，所以必须事先要把表创建好。下面的代码给出了创建表的全过程。

```
<?
require "17-1.php";                                          //调用配置文件
$link=mysql_connect($db_host,$db_user,$db_pass)or die(mysql_error());      //连接主机
mysql_select_db($db_name,$link);                             //选择数据库
$sql="create table $table_vote(
        id int(5) not null auto_increment primary key,
        v_name varchar(50) not null default '',
        v_num int(5) not null default 0,
        v_type int(1) not null default 0
)";                                                          //创建 vote 表的 SQL 语句
if(mysql_query($sql,$link))                                  //发送 SQL 请求
echo "投票表已经创建成功！<p>";                              //如果 SQL 语句成功执行，则显示信息
else
echo "创建表时出现错误，表未被成功创建。<p>";
$sql="create table $table_record(
        id int(5) not null auto_increment primary key,
        r_id int(5) not null default 0,
        r_name varchar(50) not null default '',
        r_num int(5) not null default 0,
        r_color int(1) not null default 0
)";                                                          //创建 record 表的 SQL 语句
if (mysql_query($sql,$link))                                 //发送 SQL 请求并做判断
echo "记录表已经创建成功！<p>";                              //如果 SQL 语句被成功执行，则显示创建成功信息
else                                                         //如果创建表失败，则显示信息
echo "创建记录表时出错，记录表未被成功创建。<p>";
?>
```

把以上代码保存为“17-2.php”，然后在 PHP 执行环境下执行该 PHP 文件，执行结果将会如图 17.1 所示。

从图 17.1 的执行结果的提示可以看出，相应的表已经被正确创建。随着表的创建完成，编写投票程序的准备工作就完成了。从 17.4.2 小节开始就来创建投票系统的各个子模块。

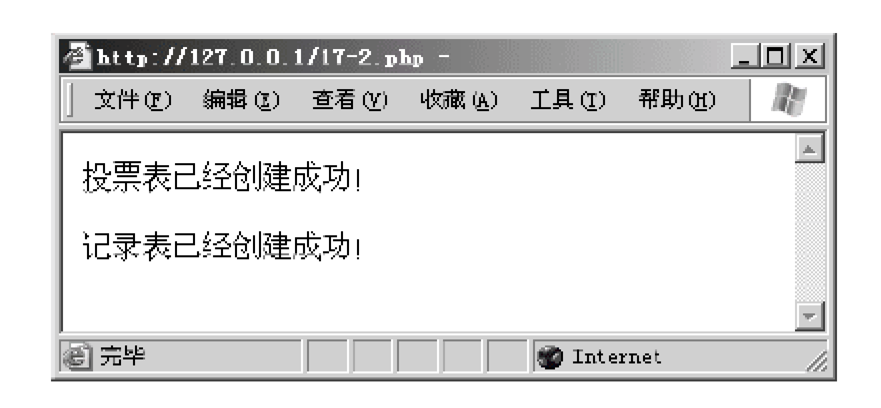


图 17.1　创建数据表执行结果

17.4.2　创建显示所有投票项的页面

本节来编写显示所有投票项的页面。该页面的作用是把所有已经存在的投票项显示给管理者，以便于用户对这些投票项进行相应的管理。

下面为显示所有投票项页面的代码：

```
<html>
<head>
<title>投票程序——显示所有投票项</title>
</head>
<body>
<center>
<h1>投票程序——显示所有投票项</h1>
<p>
<a href=17-4.php>添加记录</a>
<p>
<?
require "17-1.php";
$link=mysql_connect($db_host,$db_user,$db_pass)or die(mysql_error());
mysql_select_db($db_name,$link);                    //连接主机选择数据库
$sql="select * from $table_vote";                   //选择所有投票项记录
$result=mysql_query($sql,$link);                    //发送 SQL 请求
$rows=mysql_fetch_array($result);                   //计算总记录数
if($rows==0)                                        //如果没有记录，则显示内容
{
     echo "现在还没有记录！";
}
else                                                //如果存在记录，则显示出记录
{
     echo "<table border='1'>";
     echo "<tr>";
     echo "<td width='10%'>";
     echo "项";
     echo "</td>";
     echo "<td width='80%'>";
     echo "名称";
     echo "</td>";
     echo "<td width='10%'>";
```

```
        echo " ";
        echo "</td>";
        echo "</tr>";
        $i=0;
        $sql="select * from $table_vote";              //再次执行 SQL 请求
        $result=mysql_query($sql,$link);
        while($row=mysql_fetch_array($result))         //遍历结果
        {
            $i++;                                      //循环变量，显示数值
            echo "<tr>";
            echo "<td>";
            echo "第".$i."条";                          //显示数值
            echo "</td>";
            echo "<td>";
            echo "<a href=17-6.php?id=".$row[id].">".$row[v_name]."</a>";
            echo "</td>";
            echo "<td>";
            echo "<a href=17-5.php?id=".$row[id].">删除</a>";
            echo "</td>";
            echo "</tr>";
        }
        echo "</table>";
    }
    ?>
    </center>
    </body>
    </html>
```

把以上代码保存为“17-3.php”，以备后用。

17.4.3　创建添加投票记录页面

本节来创建管理者添加新的投票项的页面。当用户执行该页面后，该添加操作分 3 步进行。第 1 步，用户输入要创建的投票项的名称，选择投票子项的条数，并选择该投票项是多选还是单选。第 2 步，用户分别输入每一条投票子项的内容，并选择每一条子项对应的颜色。第 3 步单击“确定”按钮，完成所有的创建工作。

由于该文件整合了添加投票记录的 3 步操作，所以要对每一步要进行什么内容的操作进行判断。这样把所有操作整合起来虽然会使代码比较长，但却减少了文件的个数，更便于用户对整个系统的管理。

具体内容请参看如下代码：

```
<?
if((!$_POST[v_name]) &&(!$_POST[r][1]))          //没有任何参数为第 1 步
{
?>
<html>
```

```
<head>
<title>投票程序——增加投票项第一步</title>
</head>
<body>
<script language=javascript>
function juge(form)
{
        if (form.v_name.value == "")
        {
                alert("请输入投票项名称!");
                theForm.v_name.focus();
                return (false);
        }
}
</script>
<center>
<h1>投票程序——增加投票项第 1 步</h1>
<p>
<h3>投票项属性</h3>
<p>
<table border=1>
<form method="post" action=<?_SERVER[PHP_SELF]?> onsubmit="return juge(this)">
<tr>
<td>输入投票项内容：</td>
<td><input type=text name=v_name></td>
</tr>
<tr>
<td>选择投票项的类型</td>
<td>
<input type=radio name=v_type value=0 checked>单选
<input type=radio name=v_type value=1>多选</td>
</tr>
<tr>
<td>选择投票项的项数</td>
<td>
<select name=v_m size=1>
<option value=2>2</option>
<option value=3>3</option>
<option value=4>4</option>
<option value=5>5</option>
<option value=6>6</option>
<option value=7>7</option>
<option value=8>8</option>
<option value=9>9</option>
<option value=10>10</option>
</select>
</tr>
<tr>
<td colspan="2">
```

```
<center><input type=submit value="下一步"></center>
</td>
</tr>
</form>
</table>
</center>
</body>
</html>
<?
}
else if(!$_POST[r][1])                              //没有选择项参数为第 2 步
{
?>
<html>
<head>
<title>投票程序——增加投票项第 2 步</title>
</head>
<body>
<script language=javascript>
function juge(form)
{
        if (form.v_name.value == "")
        {
                alert("请输入投票项名称!");
                theForm.v_name.focus();
                return (false);
        }
}
</script>
<center>
<h1>投票程序——增加投票项第二步</h1>
<p>
<h3>投票项每项选择项属性</h3>
<p>
<table border=1>
<form method="post" action=<?_SERVER[PHP_SELF]?> onsubmit="return juge(this)">
<?
$v_name=$_POST[v_name];                             //获取传入变量
$v_type=$_POST[v_type];
$v_m=$_POST[v_m];
echo "<tr><td colspan=4><center>".$v_name."</center></td></tr>\n";
echo "<input type=hidden name=v_name value=".$v_name.">\n";
echo "<input type=hidden name=v_type value=".$v_type.">\n";
echo "<input type=hidden name=v_m value=".$v_m.">\n";
for($i=1;$i<($v_m+1);$i++)                          //循环显示每个选择项
{
        echo "<tr>";
        echo "<td>选择项".$i."内容：</td>\n";
        echo "<td>";
```

```
        echo "<input type=text name=r[]>";
        echo "</td>\n";
        echo "<td>";
        echo "颜色";
        echo "</td>\n";
        echo "<td>\n";
        echo "<select size=1 name=c[]>\n";
        echo "<option value=1>红</option>\n";
        echo "<option value=2>蓝</option>\n";
        echo "<option value=3>绿</option>\n";
        echo "<option value=4>黄</option>\n";
        echo "<option value=5>紫</option>\n";
        echo "</td>\n";
        echo "</tr>\n";
}
?>
<tr>
<td colspan="4">
<center>
<input type=button value="上一步" onclick=history.go(-1)>
<input type=submit value="下一步">
</center>
</td>
</tr>
</form>
</table>
</center>
</body>
</html>
<?
}
else                                        //除以上两种情况之外为第 3 步
{
        $v_name=$_POST[v_name];             //获取传入变量
        $v_type=$_POST[v_type];
        $v_m=$_POST[v_m];
        for($i=0;$i<$v_m;$i++)              //循环读取数组变量
        {
                $r[]=$_POST[r][$i];
                $c[]=$_POST[c][$i];
        }
        require "17-1.php";
        $link=mysql_connect($db_host,$db_user,$db_pass);
        mysql_select_db($db_name,$link);    //连接服务器选择库
        $sql="insert into $table_vote(v_name,v_type) values('$v_name','$v_type')";
        mysql_query($sql,$link);            //插入投票记录
        $sql="select max(id) from $table_vote";
        $result=mysql_query($sql,$link);
        $row=mysql_fetch_array($result);
```

```
        for($i=0;$i<$v_m;$i++)                          //循环插入选择项记录
        {
            $temp=$r[$i];
            $temp2=$c[$i];
            $sql="insert into $table_record(r_id,r_name,r_color) values('$row[0]','$temp','$temp2')";
            mysql_query($sql,$link);                    //插入记录
        }
?>
<html>
<head>
<meta http-equiv="refresh" content="2; url=17-3.php">
<title>投票程序——增加投票项第三步</title>
</head>
<body>
投票项创建成功！<p>
两秒后返回
</body>
<?
}
?>
```

把以上代码保存为“17-4.php”，以备后用。

17.4.4　创建删除投票项的页面

本节创建删除投票项的页面。该文件首先要获取用户提交的 ID，然后从表中选择出相应的 ID，并进行删除操作。删除投票的实质，要对两个表进行操作。一要删除表 vote 中指定 ID 的记录，另一方面也要删除表 record 中 r_id 等于指定 ID 的记录，也就是删除投票项相应的选择支。

下面就来具体创建该页面。内容请参看以下代码：

```
<?
if(!$_GET[id])                                      //如果没有指定删除 ID，则显示信息
{
    echo "没有指定 ID！";
    exit();
}
else                                                //如果指定了删除 ID，则执行操作
{
require "17-1.php";                                 //调用配置文件
$link=mysql_connect($db_host,$db_user,$db_pass)or die(mysql_error());      //连接主机
mysql_select_db($db_name,$link);                    //选择数据库
$sql="delete from $table_vote where id='$_GET[id]'";    //删除指定投票项
if(mysql_query($sql,$link));                        //发送 SQL 请求并对结果进行判断
{
    echo "成功删除投票项！";
}
else
{
```

```
        echo "删除投票项时出现错误！";
}
echo "<p>";
$sql2="delete from $table_record where r_id='$_GET[id]'";//删除指定选择项，即某个投票项的所有选择支
if(mysql_query($sql2,$link));                              //发送 SQL 请求并判断结果
{
        echo "成项删除选择项！";
}
else
{
        echo "删除选择项时出现错误！";
}
echo "<html>";
echo "<head>";
echo "<title>投票程序——删除投票项</title>";
echo "<meta http-equiv=\"refresh\" content=\"2; url=17-3.php\">";
echo "</head>";
echo "<body>";
echo "成功删除投票项记录！";
echo "<p>";
echo "两秒后返回";
echo "</body>";
?>
```

将上面的代码保存为“17-5.php”，以备后用。

17.4.5　创建显示投票项页面

本节就编写投票项的显示页面。注意本节与 17.4.2 节的不同。17.4.2 节显示所有投票项，即本节显示单一投票项的详细情况，即每一个投票项和它所包括的选择项。

这里采用两种方式来显示，一种是采用通常的页面方式显示，另一种是采用 JavaScript 方式来显示。为什么这样呢？因为通常网上的普通空间并不支持 PHP，而采用 JavaScript 方式显示只需提供用相应的 JS 代码就可以实现投票功能了。而普通的显示方式，则是提供给支持 PHP 代码的用户，因为这些用户完全可以直接拿来使用。

先来编写普通的显示某一条投票项的页面。代码如下：

```
<?
if(!$_GET[id])                                             //如果没有指定 ID 号
{
        echo "没有指定 ID！";
        exit();
}
else
{
echo "<html>";
echo "<head>";
echo "<title>投票程序——显示投票项</title>";
```

```
echo "</head>";
echo "<body>";
echo "<center>";
require "17-1.php";                                         //调用配置文件
$link=mysql_connect($db_host,$db_user,$db_pass)or die(mysql_error());
mysql_select_db($db_name,$link);                           //选择数据库
$sql="select * from $table_vote where id='$_GET[id]'";      //显示指定投票项
$result=mysql_query($sql,$link);                            //发送 SQL 请求
$row=mysql_fetch_array($result);                            //计算总列数
$s=$row[v_num];                                             //赋值给变量
echo "<h1>$row[v_name]</h1>";                               //显示投票项标题
echo "<p>";
echo "<table border=\"1\" width=60%>";
echo "<form action=17-9.php method=post>";
echo "<input type=hidden name=id value=".$row[id].">";
echo "<input type=hidden name=v_type value=".$row[v_type].">";
echo "<tr>";
echo "<td> </td><td>选项</td><td>被选择情况</td>";
echo "</tr>";
$sql2="select * from $table_record where r_id=$_GET[id]";
$result2=mysql_query($sql2,$link);                          //发送 SQL 请求以显示选择项
while($rows=mysql_fetch_array($result2))
{
    echo "<tr>";
    echo "<td>";
    if($row[v_type]==0)                                     //如果选择类型是单选
    {
        echo "<input type=radio name=r value=".$rows[id].">";
    }
    else                                                    //如果选择类型是多选
    {
        echo "<input type=checkbox name=r[] value=".$rows[id].">";
    }
    echo "</td>";
    echo "<td>".$rows[r_name]."</td>";
    echo "<td>";
    if($rows[r_num]==0)                                     //如果被投票数为 0
    $width=0;                                               //图片宽度为 0
    else
    $width=$rows[r_num]/$s;                                 //图片宽度为得票数除以总投票数
    if($width!=0)                                           //如果被投票数不为 0 显示图片
    {
        echo "<img src=".$rows[r_color].".bmp width=".($width*200)." height=10>\n";
    }
    echo $rows[r_num]."/".$row[v_num];                      //显示数字
    echo "</td>";
    echo "</tr>";
}
echo "<tr><td colspan=3><center><input type=submit value=\"确认提交\"></center></td></tr>";
```

```
echo "</form>";
echo "</table>";
}
?>
```

将以上代码保存为“17-6.php”，以备后用。

下面再来创建一个调用 JS 显示相应内容的 PHP 文件。具体内容参看以下代码：

```
document.write("<? require '17-1.php';
$link=mysql_connect("$db_host","$db_user","$db_pass");
mysql_select_db($db_name,$link);
$sql="select * from $table_vote where id='$_GET[id]'";
$result=mysql_query($sql,$link);
$row=mysql_fetch_array($result);
$s=$row[v_num];
echo "<center>";
echo '<h1>'.$row[v_name].'</h1>\n';
echo "<p>";
echo '<table border=1 width=60%>\n';
echo '<form action=17-9.php method=post>\n';
echo '<input type=hidden name=id value='.$row[id].'>\n';
echo '<input type=hidden name=v_type value='.$row[v_type].'>\n';
echo '<tr>';
echo '<td> </td><td>选项</td><td>被选择情况</td>\n';
echo '</tr>';
$sql2="select * from $table_record where r_id=$_GET[id]";
$result2=mysql_query($sql2,$link);
while($rows=mysql_fetch_array($result2))
{
    echo '<tr>';
    echo '<td>';
    if($row[v_type]==0)
    {
        echo '<input type=radio name=r value='.$rows[id].'>\n';
    }
    else
    {
        echo '<input type=checkbox name=r[] value='.$rows[id].'>\n';
    }
    echo '</td>\n';
    echo '<td>'.$rows[r_name].'</td>\n';
    echo '<td>\n';
    if($rows[r_num]==0)
    $width=0;
    else
    $width=$rows[r_num]/$s;
    if($width!=0)
    {
        echo '<img src='.$rows[r_color].'.bmp width='.($width*200).' height=10>\n';
    }
```

```
        echo $rows[r_num].'/'.$row[v_num];
        echo '</td>\n';
        echo '</tr>\n';
    }
    echo '<tr><td colspan=3><center><input type=submit value=确认提交></center></td></tr>\n';
    echo '</form>\n';
    echo '</table>\n';
    echo '</center>';
    ?>
    ");
```

把以上代码保存为“17-7.php”。

下面是调用该 JS 页（17-7.php）的普通 HTML 页面：

```
<html>
<head>
<title>投票程序——调用 JS 页</title>
</head>
<body>
<script language=javascript src="http://127.0.0.1/17-7.php?id=1">
</script>
</body>
</html>
```

把以上代码保存为“17-8.php”。由于该页面中没有任何可执行的 PHP 代码，所以该页面可以是普通的 HTML 页。通过以上代码可见，要调用 JS 页时要使用

```
<script language=javascript src="http://127.0.0.1/17-7.php?id=1">
</script>
```

样式，其中的 src 为 17-7.php 的相对或者绝对路径，注意不能省略 ID 值。这样可以在不支持 PHP 的网站上调用该投票程序。当然，如果是提供给普通用户使用，其中的 src 就一定要是绝对路径，因为用户的内容不可能和服务器上的程序在一个路径之下。

另外，单独使用这样的调用即显示投票情况是会成功的。但要处理这样的提交结果，就一定要把下面要讲到的提交处理页面 17-9.php 中最后回退的页面改为用户的页面，这就需要程序在处理时使用系统变量$_SERVER['HTTP_REFERER']记录用户上一页的内容，然后在处理结束后重新返回用户的页面。

17.4.6　创建选择项提交处理页面

本节编写对选择项的提交进行处理的页面。该页面实现的功能是依照用户选择的项目更新相应记录的数值。具体内容参看以下代码：

```
<?
$id=$_POST[id];                                   //获取用户提交的 ID 值
$v_type=$_POST[v_type];                           //获取投票项的类型
$r=$_POST[r];                                     //获取投票项内容
```

```
require "17-1.php";
$link=mysql_connect($db_host,$db_user,$db_pass);
mysql_select_db($db_name,$link);
if($v_type==0)                                                          //如果投票类型是单选
{
    $sql="update $table_vote set v_num=v_num+1 where id=$id";           //更新投票记录加 1
    mysql_query($sql,$link);
    $sql2="update $table_record set r_num=r_num+1 where id=$r";         //更新选择支记录加 1
    mysql_query($sql2,$link);
}
else                                                                    //如果投票项是多选
{
    for($i=0;$i<count($r);$i++)                                         //通过循环实现记录的更新
    {
            $sql="update $table_vote set v_num=v_num+1 where id=$id";
            mysql_query($sql,$link);
            $temp=$r[$i];
            $sql2="update $table_record set r_num=r_num+1 where id=$temp";
            mysql_query($sql2,$link);
    }
}
echo "<html>";                                                          //处理结束显示 HTML 内容
echo "<head>";
echo "<title>投票程序——处理投票结果</title>";
echo "<meta http-equiv=\"refresh\" content=\"2; url=17-6.php?id=$id\">";    //返回 17-6.php。
echo "</head>";
echo "<body>";
echo "投票成功！";
echo "<p>";
echo "两秒后返回";
echo "</body>";
?>
```

把以上代码保存为“17-9.php”。至此，该投票程序的全部主要功能已经完成。17.5 节开始对所有功能进行调试。

17.5　测 试 程 序

程序测试是程序编写完成后的一个重要环节。每个程序在编写完成后都要进行测试。通过对代码的测试，可以发现其中的 BUG，然后就可以排除 BUG，使程序更好地运行。

在 17.4 节中已经完成了投票程序的全部模块，本节就对 17.4 节所编写的所有页面进行测试，以验证该程序是否能正常运行。操作步骤如下：

（1）由于准备工作（配置文件、创建表的工作）在 17.4.1 小节中已经完成，所以本节就直接从显示投票项开始。第 1 步，在 PHP 执行环境中运行 17-3.php，将显示如图 17.2 所示的结果。因为，虽然已经创建了相应的表，但是表中并没有任何记录，所以图 17.2 才显示现在还没有记录的结果。这时没

有别的选择，只有单击链接“添加记录”，来添加投票记录，即执行 17-4.php。其执行结果如图 17.3 所示。

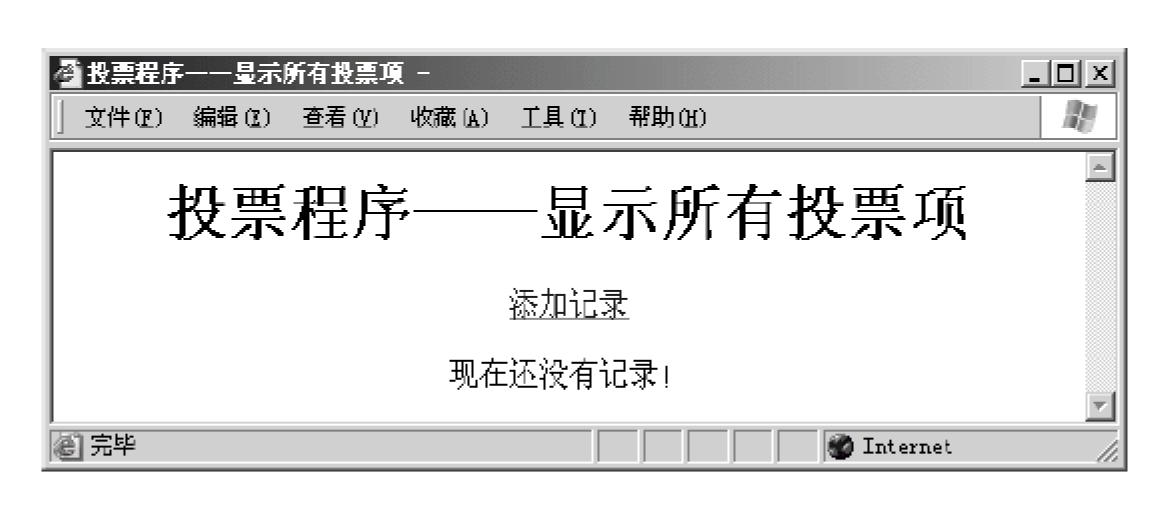


图 17.2　第一次运行显示投票项页面时的结果

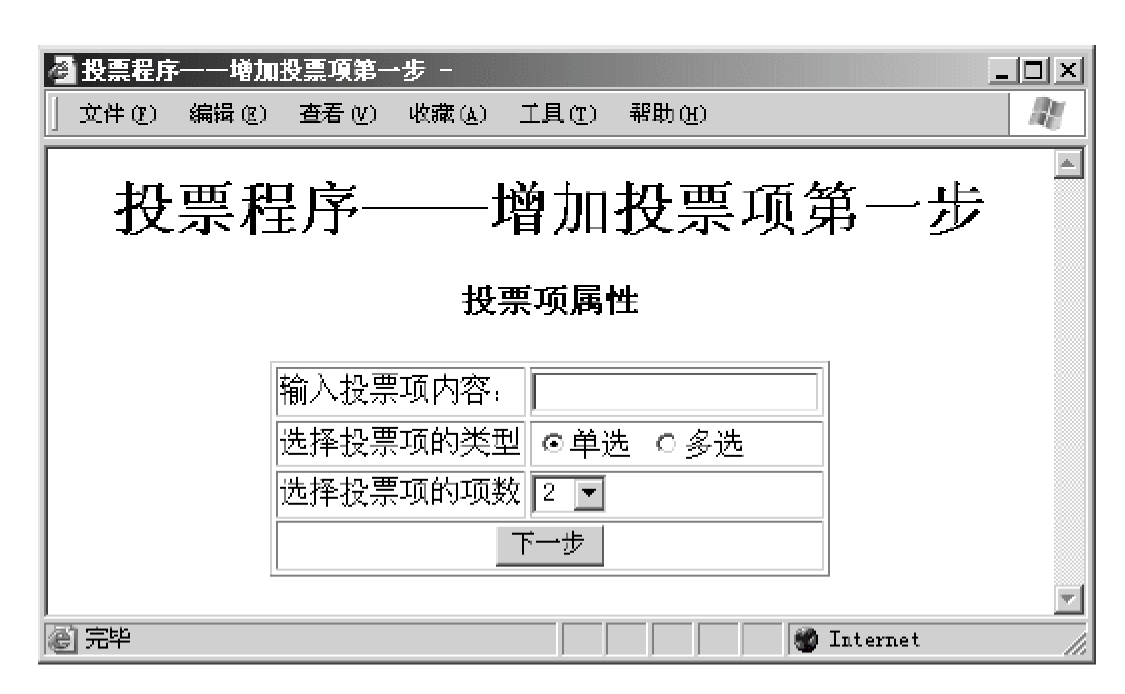


图 17.3　增加投票项第一步执行结果

（2）由于 17-4.php 把增加投票项的 3 步操作化为了一步，所以第 1 步操作执行时，将会发现如图 17.3 所示的提示。下面开始按需要创建新的投票项。这里投票项的内容定为：“你对本站印象如何”；投票项的类型为单选项；投票项的项数选择 3 项。然后，单击“下一步”按钮，就会转到增加投票项的第 2 步。第 2 步的显示结果如图 17.4 所示。

（3）由于在上一步选择了 3 项选择项，所以这一步就是确定每项选择项的内容。这里分别填入好极了、很一般、糟透了 3 项内容。颜色分别选用：红、黄、蓝 3 种颜色。按照要求填写完成相应项目后，再单击“下一步”按钮，就转到了增加投票项的第 3 步。由于第 3 步的操作是把相应记录添加到表中，所以执行都是在后台运行。执行完毕后，直接跳转到了“显示所有投票项”的页面，即 17-3.php。跳转后的结果如图 17.5 所示。

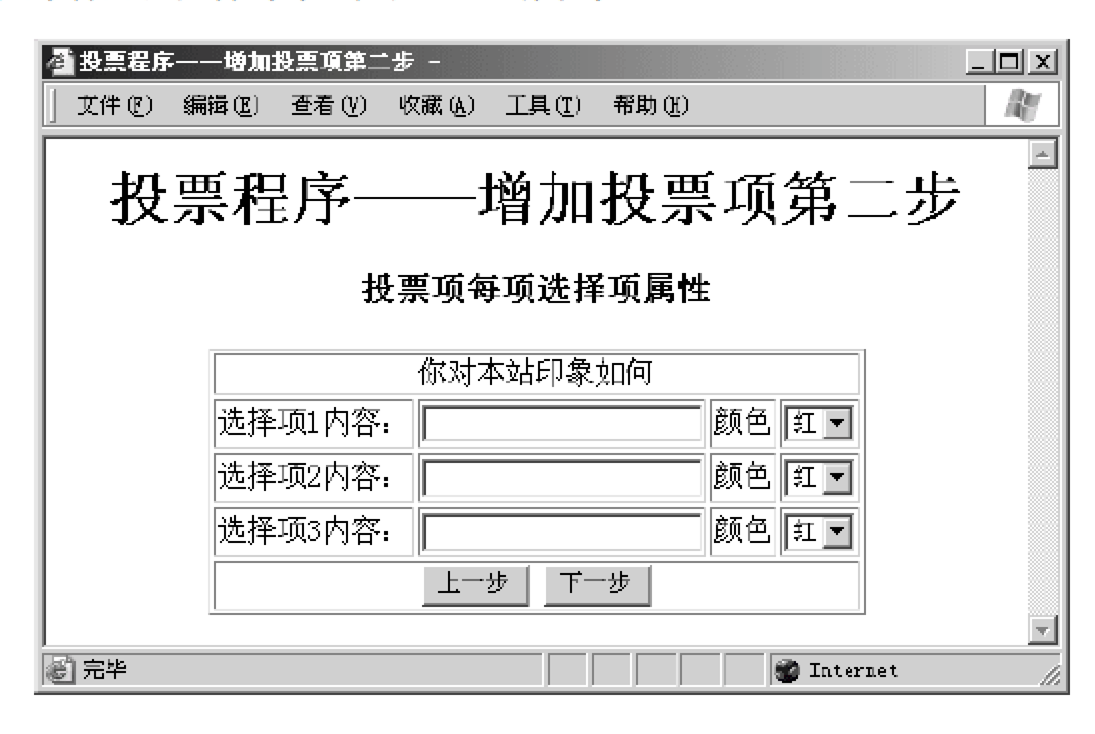


图 17.4　增加投票项第二步执行结果

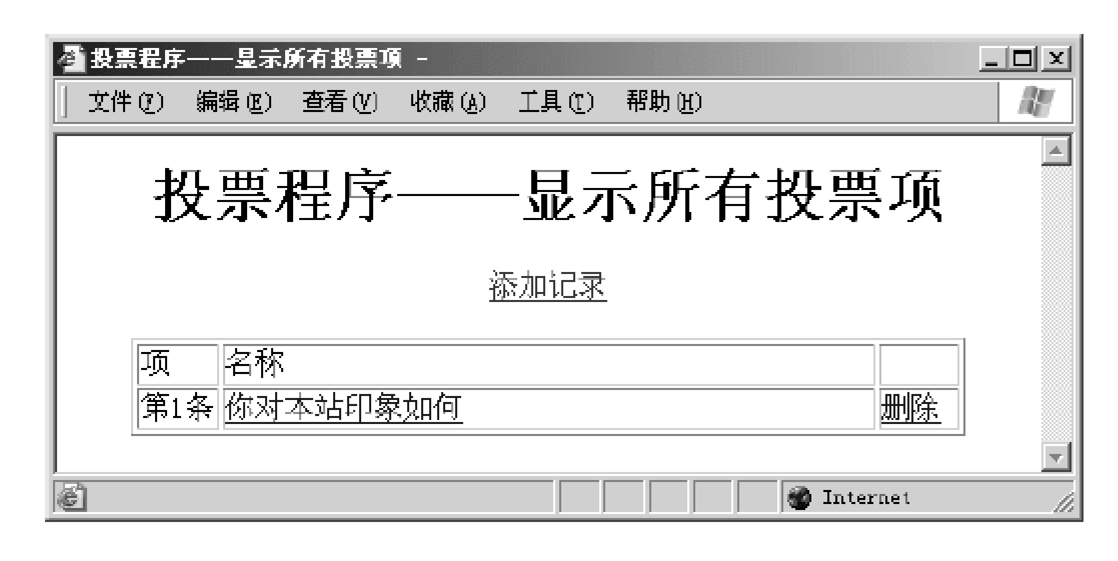
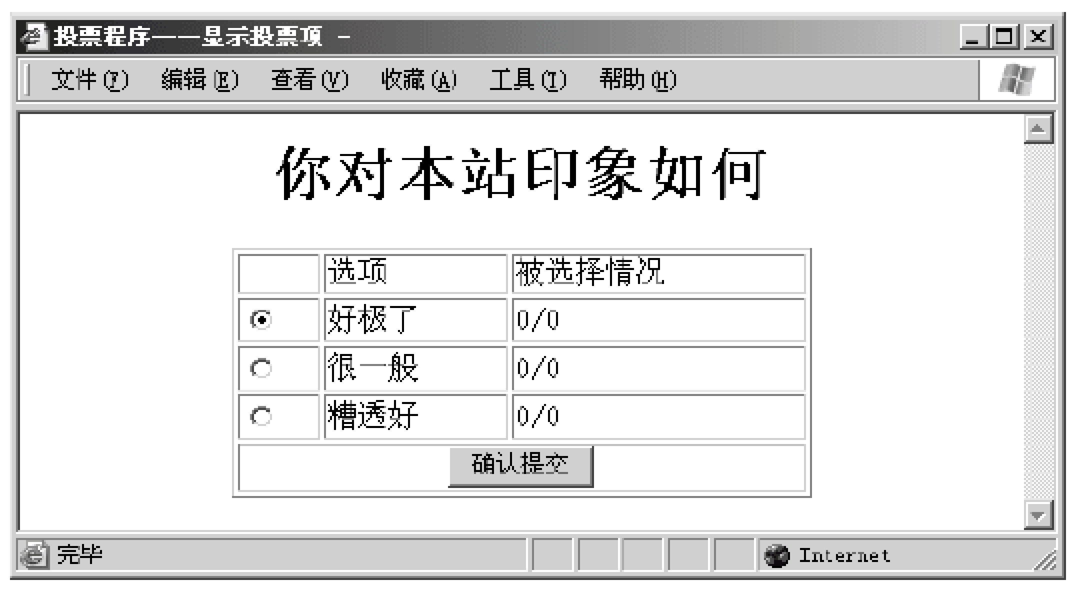


图 17.5　添加完投票项后的显示投票项页面

（4）通过图 17.5 的执行结果可以看到，已经正常显示出了前面添加过的投票项“你对本站印象如何”。这说明程序正常运行。单击投票项名称的链接，将会转到显示投票项的页面，即相当于执行 17-6.php。执行结果如图 17.6 所示。

（5）从图 17.6 可以看出，投票程序正确地显示了投票项及其选择项的内容。由于此时并没有任何的票，或者说没有做任何的选择，所以每一个选择项的选择次数都是 0 次。下面尝试选择图 17.6 中的“好极了”，然后确认提交。由于该页表单的 action 属性指向 17-9.php，所以将会调用 17-9.php 来处理提交结果。

（6）像上面添加记录一样处理页面更新表中的内容，即把 vote 中的 v_num 增加 1，再把 record 中的相应项的 r_num 增加 1。处理完成后，将返回显示投票项详细信息的页面。处理结果如图 17.7 所示。



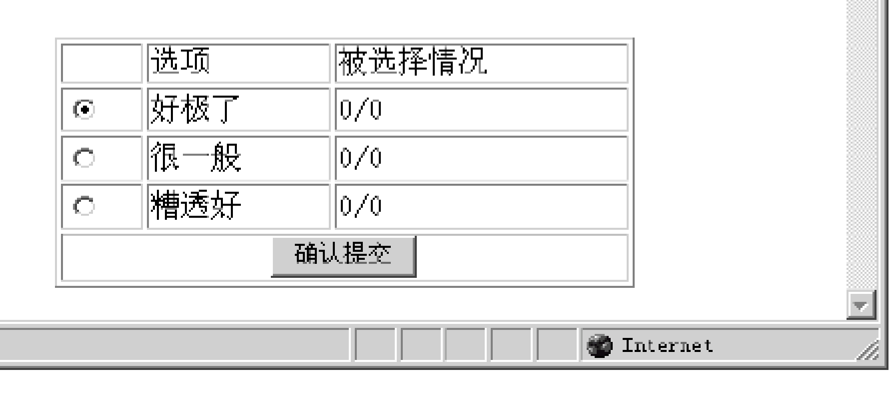
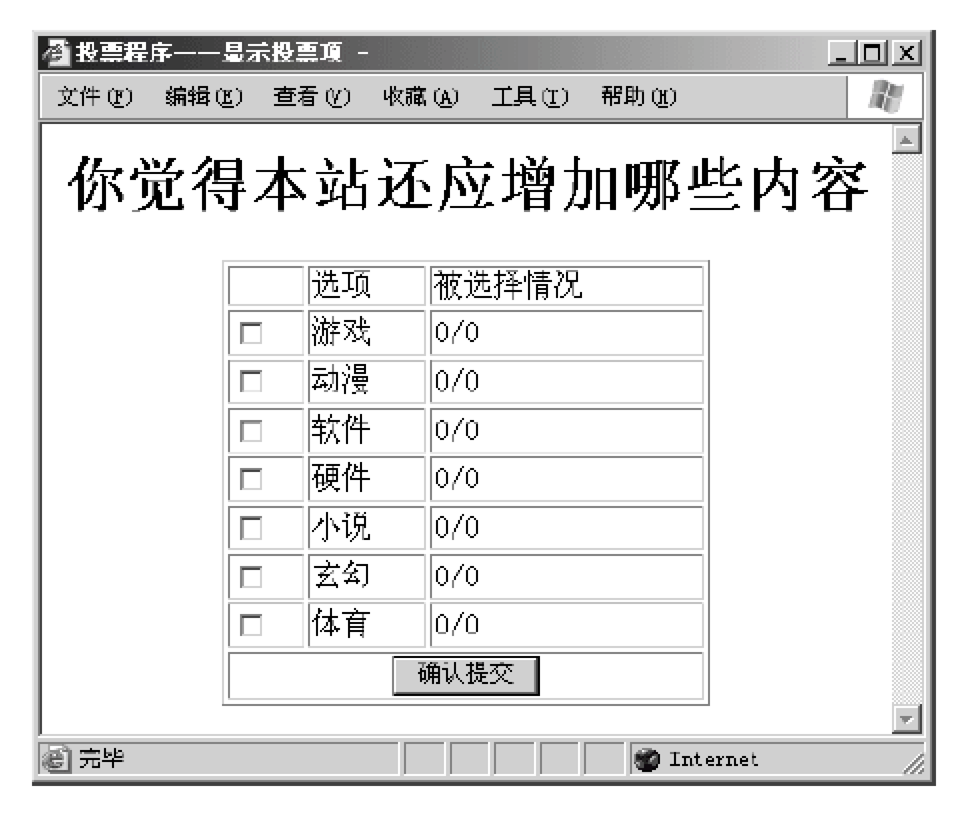


图 17.6　显示投票项详细信息页面　　图 17.7　显示投票项的投票结果

每一项投票结果都被正确地显示了出来。说明该程序正确处理了用户提交的投票结果。

（7）本步骤测试一下多项选择是否能正确执行。先运行 17-4.php 创建一个多项投票项，如标题为：“你觉得本站还应增加哪些内容”；选择项为“游戏”、“动漫”、“软件”、“硬件”、“小说”、“玄幻”、“体育”。把每一项都设定相应的颜色，创建好以后的显示信息如图 17.8 所示。

（8）下面分别一次选择其中的多项内容，以测试多选项是否成功执行。执行结果如图 17.9 所示。



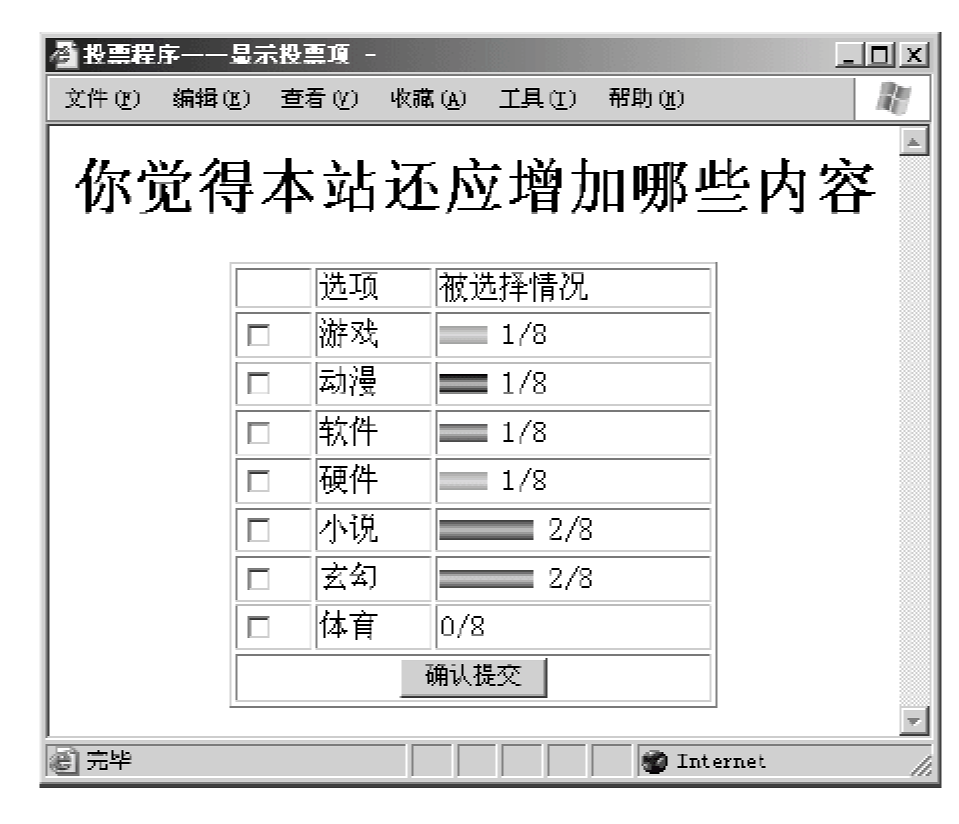



图 17.8　显示多选投票项详细情况　　图 17.9　显示多项投票项投票结果

（9）如图 17.9 所示，如果用户一次选择 4 项，就会当作单选 4 次的投票被计入库中。而每被投中的一次将会作为一次投票。经测试达到了正确处理多项选择的目的，即把一次多选的几项当作几次计数，如一次多选了多项内容，就当作多次计数。

（10）重新回到 17-3.php，测试删除投票项的功能。在执行删除操作之前，共有两条投票项（即测试时添加的两条），如图 17.10 所示。

（11）单击其中一条后面对应的删除链接，将会执行 17-9.php。该页面的功能是把参数 ID 所对应的投票项及其在 record 表中的所有选择项全部删除。在执行完全部操作后，又会自动跳回到 17-3.php。所以只有通过操作前后的对比，才能发现程序是否正确执行。其结果如图 17.11 所示。

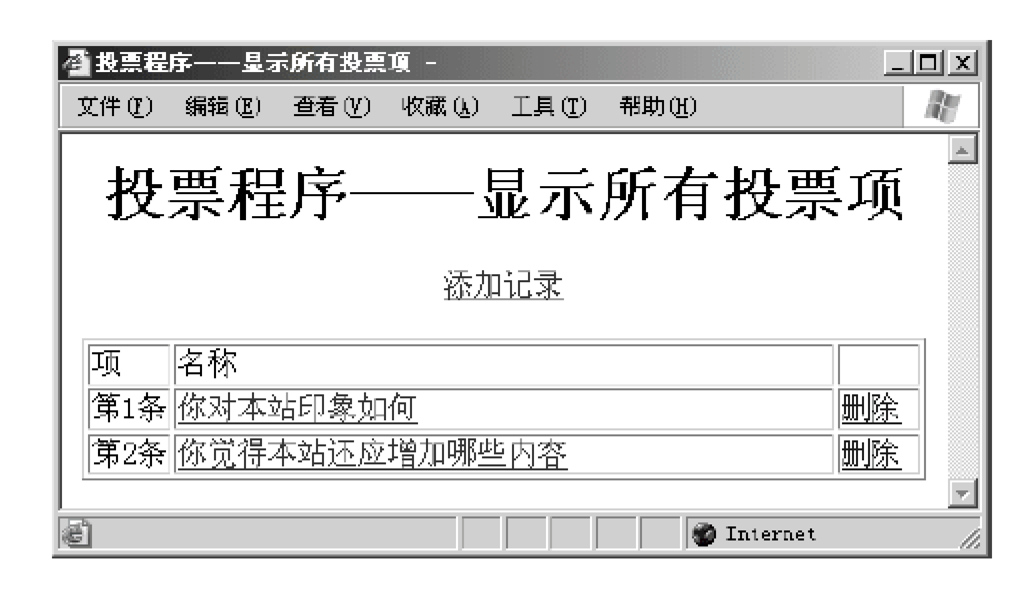



图 17.10　删除前的所有投票项

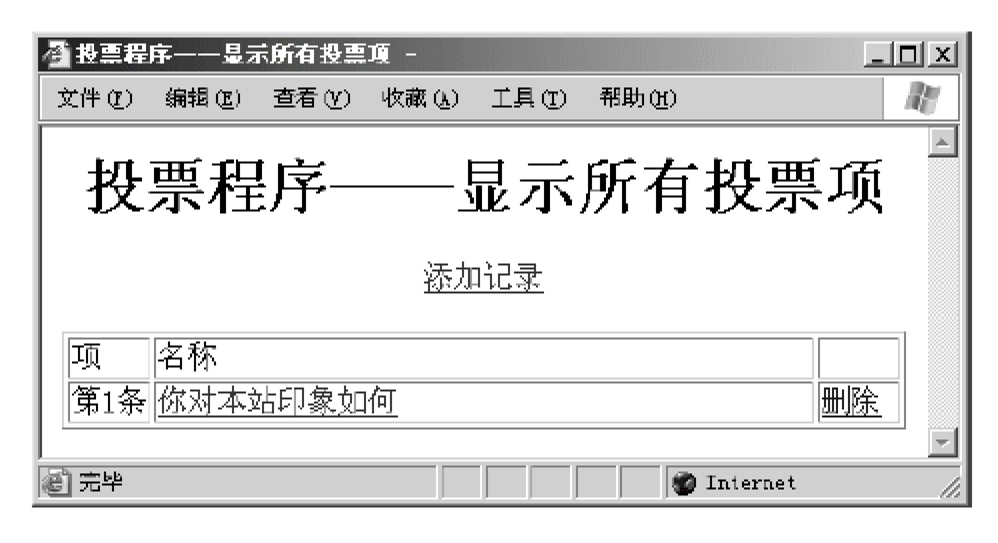


图 17.11　删除后的所有投票项

（12）通过对图 17.10 与 17.11 的对比可以发现第 2 条记录不见了，表示它已经被成功删除了。

（13）下面测试 JS 调用页面的功能。要测试这项功能，直接在浏览器地址栏中运行 17-8.php。其结果如图 17.12 所示。

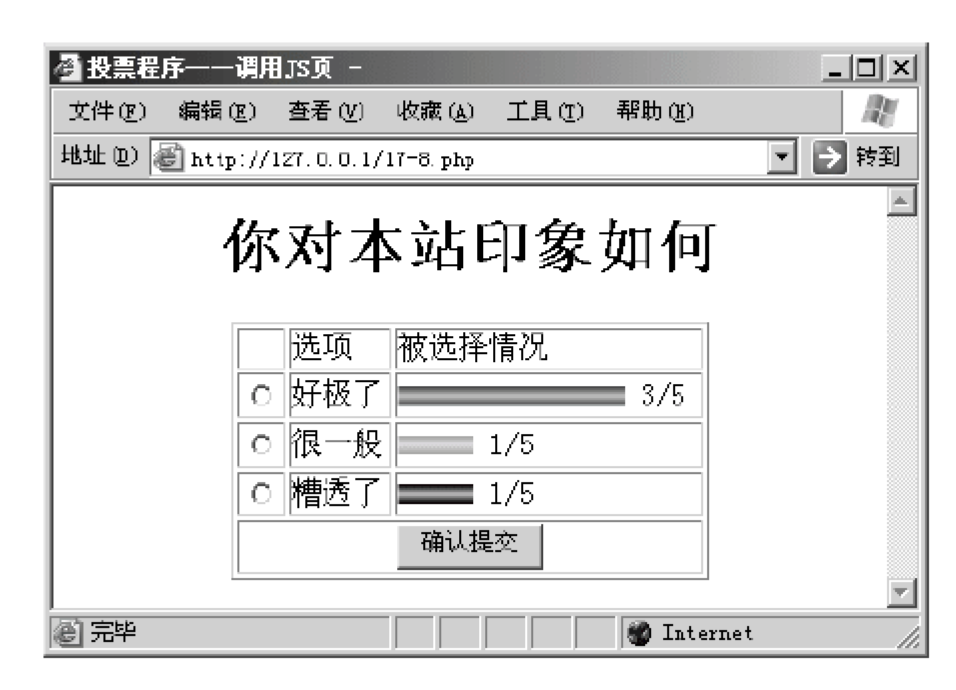


图 17.12　使用 JS 调用页面的执行结果

（14）通过图 17.12 可以看出，通过 JS 调用，显示出了与直接使用显示投票项页面几乎完全一致的内容。说明该功能也正常。需要注意的是，当用户使用 JS 调用的页面，提交时必须要把 17-9.php 处理操作结束后所要回转的页面地址做一下改动。使用系统变量$_SERVER['HTTP_REFERER']，重新返回用户的页面。

至此，该投票系统的所有功能全部测试完毕。

17.6　如何防止重复投票

投票程序该具有的功能已经全部都有了，但还是不很完善，因为它不能限制用户重复投票。这里给出一个改进思路。由于实现起来相当简单，对于学过本书的读者已经完全有能力自己解决，所以这里也就不再给出代码，有兴趣的读者可以自己通过实践来解决。

还记得 17-1.php 那个$re_vote（重复投票）变量吗？在执行 17-9.php 处理用户提交操作之前，先来判断该值，如果该值设定为 True，即允许用户重复投票，就不执行任何操作。如果该值为 False，则先判断有无某个 Cookie 值，如果没有，把用户 IP 写入某个 Cookie 值（如何获得用户 IP，请参阅本书第 9 章的内容）。如果存在该 cookie 值，则把用户的 IP 与该值相比较，如果二者不一致就继续执行下面的操作，如果二者一致就给出不允许重复投票的提示，并且使用 exit()函数中止所有 PHP 语句的执行。

这样一来就可以有效防止用户的重复投票，这样既增加了投票结果的真实性、可信度又可以减轻服务器的负担。

17.7　本 章 小 结

本章完成了一个网上投票程序。通过对本章的学习，一方面复习了数据库的知识，另一方面也使读者了解到了如何采用不同的方法来解决相同的问题，进一步增强了读者使用 PHP 来处理实际问题的能力。但是，本章所举的投票程序的例子毕竟还只是十分简单的一个实例，实际应用时投票程序往往需要有多方面的要求。但不管它们的功能有多么丰富，内容有多么复杂，其最核心的内容还是对数据库表的操作。相信读者通过本章的学习，加上自己的思考与努力，一定能写出高效、简洁的网上投票程序。

第18章

文本留言板程序

- 留言板的实现方法
- 文本留言板的实现原理
- 文本文件的读取操作
- 文本文件的写入操作
- 文本文件的综合处理

留言板在网络上有着极为广泛的应用，它是沟通网站管理人员与普通浏览者的一个强有力的纽带。不论是个人的小型网站，还是企业的大型站点，到处都可以发现留言板的影子。如果一个网站中没有了留言板，那就无法和普通浏览的用户进行有效的沟通。因为留言板有如此广泛的应用，所以，编写留言板，就是编写大型Web应用程序的基础。

这一章将向读者介绍如何编写一个文本型的留言板。本章内容包括以下几个部分，留言板分类、文本留言板的实现原理、留言板实例的构成、留言板配置文件的建立、编写显示留言板及提交留言页、后台修改留言功能的实现、后台删除留言功能的实现以及进一步完善等内容。通过本章的学习，会使读者掌握如何编写PHP文本留言板。

18.1　留言板分类

留言板是网络上使用最广的互动 Web 应用程序之一。由于其使用灵活、方法简单、用户与管理者之间互动效果明显，所以一直深受网络管理者与使用者的欢迎。网上常见的留言板按照它们的存储机制的不同，大致可以分为以下两大类：文本型留言板和数据库型留言板。这一节为读者分别作介绍。

18.1.1　文本型留言板

顾名思义，所谓文本型留言板就是采用文本文件作为数据存储方式的一类留言板。文本留言板由于采用文本文件作为数据存储的载体，所以与其他类型的留言板相比较，具有几下几个特征。

- 由于不需要数据库支持，所以文本留言板只要服务提供商支持普通 PHP 程序，就可以使用。
- 与其他类型留言板程序相比，可移植性强。数据备份也更为简单，只需要将相应数据文件备份即可。
- 文本留言板在处理内容比较少的数据时往往速度比较快，但是在处理大量数据，如留言数据达到成千上万的数量级时就会显得力不从心。
- 由于没有专门的函数，所以与数据库留言本相比，插入、修改、删除记录操作起来相对要麻烦一些。
- 由于文本留言板的数据采用纯文本的方式进行排列，所以记录的检索起来也不如数据库留言板那么方便。

以上列出了文本留言板的几个特点，可见与其他类型的留言板相比起来，也有不足之处。但它的优点也是相当明显的，所以，文本留言板还是在网络上有相当多的拥护者。这也是这一章所要介绍的重点。

18.1.2　数据库型留言板

文本型留言板固然有很多优点，但不足之处也是显而易见的。如不具备搜索功能；插入、修改、删除留言相对比较麻烦等。这些不足，在数据库型留言板中可以非常轻松地解决。由于数据库型留言板采用数据库作为数据载体，所以插入、修改、删除留言时只需执行相应的 SQL 语句即可。而留言搜索更是可以轻松实现。下面来了解一下数据库型留言板的特点。

- 数据库留言板是以数据库、数据表作为留言数据的存储载体的一类留言板程序的统称。用数据库作为数据载体，相对于文本型留言板来说，它的优点非常明显。但与之相比，缺点也是显而易见的。
- 使用数据库作为载体，在处理留言信息时速度很快，特别是在处理大量数据时，速度优势更为明显。
- 有 PHP 数据库函数库的支持，依托高效 SQL 的语句，数据库型留言板实现数据插入、修改、删除等操作都非常方便快捷。
- 留言记录搜索功能对于数据库型留言板来说更是小菜一碟。甚至它可以支持任何字段的模糊

查找。

- 由于需要数据库支持，所以对服务提供商要求较多，一旦服务供应商不支持这项服务，那么这种类型的留言板将无法运行。
- 可移植性与文本留言板比较起来较差。在一个空间运行正常的程序到了另一个地方，就要修改相关配置选项。而文本留言板几乎可以不做任何修改即可以在不同的空间上正常运行。
- 相对文本留言板来说，数据备份相对麻烦。虽然有高效的数据库管理程序，如 PHPMYADMIN 之类的可以轻松实现数据库的导出导入，但对于只使用复制、粘贴就可以轻松备份的文本留言板来说还是显得有些麻烦。

以上列出了数据库留言板的一些特点。通过对比可以发现，两种留言板互有长短，互为补充，不能说哪种比哪种更优秀，要看用在什么环境下。

如果是个人用户，推荐使用文本型留言板，因为该种类型留言板小巧玲珑，管理方便。在留言数量相对少的情况下使用者能体会到文本留言板的快捷与便利。如果是企业级的用户，还是推荐使用数据库型留言板。对于企业服务器来说，数据库支持不是问题。另外企业也不可能经常更换服务器或者空间，可移植性与数据备份的麻烦可以忽略不计。最主要的是使用数据库型留言板，可以使用数据库的强大功能，并且数据存储量可以不受限制。

这一节，留言板的类型就介绍到这里，从 18.2 节开始就来为读者介绍如何具体实现 PHP 文本留言板。

18.2　文本留言板的实现原理

这一节将介绍一下文本留言板的实现原理。

文本留言板，把用户输入的数据存入一个文本文件。虽说是文本文件，但这个文件的扩展名不一定是 txt。它可以是任意的扩展名，如 php、dat、gif、mp3、jpg、rm，甚至可以不加任何扩展名。用户输入数据的存储格式一般是一条记录占用一行，每一行都有一个行标，以及所输入的各项内容。各项内容之间使用特殊符号将它们分隔开来。特殊符号，可以自由定义，但是为了区别于普通输入内容，一般不采用标准的字符，而是用一些通常不常用到的字符，如“||”、“@@”、“~~”、“***”、“$$$”等。如下面这一段就显示了一行文本留言板的记录：

```
15||JACK||jack@hotmail.com||欢迎大家来到这里并留言！<br>我祝大家心情愉快，工作顺利！||2006 年 10 月 5 日 11:42||127.0.0.1
```

通过查看上面的内容可以发现，该文本留言板程序使用了“||”来分割用户所输入的各项内容。各项内容的含义大致如下所示：

```
15                        行号
JACK                      姓名
jack@hotmail.com          电子信箱
欢迎大家来……              留言内容
2006 年 10 月 5 日 11:42  留言日期及时间
127.0.0.1                 IP 地址
```

将留言内容按特定格式存入文本文件，在需要显示时，再按照相应的格式显示出来即可。

另外一点就是文本留言板的分页显示。它不像数据库那样读取相应的记录即可。文本留言板分页的实现原理是读取相应的行。如一共有 25 条记录，共分 25 行。如果是第 1 页，就显示第 25 到第 16 条记录；如果是第 2 页就显示第 15 到第 6 条记录；如果是第 3 页就显示第 5 到第 1 条记录，依次类推。这样就可以实现多页的显示了。

还有就是修改与删除留言数据，与数据库留言板相比，也是有点麻烦。需要对相应的记录读出，后按要求进行修改或者删除操作。

这一节，关于文本留言板的基本实现原理就讲到这里。

18.3　文本留言板实例的组成

现介绍本章要讲到的文本留言板实例的组成。

下面具体列举一下留言板系统的组成，及每个文件的作用。

将要讲解到的文本留言板程序由以下文件组成。

- 15-1.php，是系统的配置文件，里面放置几个重要的全局变量。包括文件名、管理者密码、每页显示的留言条数等内容。
- 15-2.php，是系统的最主要的文件。该文件把留言显示、留言的提交前台、留言的提交后台处理 3 项功能整合到一个文件之中。这样做既减少了文件个数，又便于管理。
- 15-3.php，是系统管理的登录页。通过该文件，管理者可以登录到服务器然后注册 Cookie。然后就可以对留言执行修改、删除等操作。
- 15-4.php，是系统的修改留言处理页面。在执行修改操作前会先判断 Cookie 值，如果 Cookie 值与系统配置文件（15-1.php）中给定的不一致，返回错误提示。如果正确就会执行修改操作。
- 15-5.php，系统的删除留言处理页面。在执行删除操作前会先判断 Cookie 值，如果 Cookie 值与系统配置文件（15-1.php）中给定的不一致，返回错误提示。如果正确就执行删除操作。

该文本留言板程序，就由以上几个文件组成。从 18.4 节开始，逐个来为读者做介绍。

18.4　文本留言板代码的实现

前面几节讲到了留言板的分类、文本留言板的实现原理以及要讲到的文本留言板的构成。这些都是为本节要讲到的内容做准备。这一节，将一个模块一个模块地实现整个系统的功能。

18.4.1　配置文件的建立

这一小节，带领读者来创建文本留言板系统的配置文件。配置文件在程序运行时起关键作用。因为该文件定义了几个对系统运行起关键作用的全局变量。它决定了系统所使用的存储数据的文件名、管理者对留言进行管理（修改、删除操作）登录时所使用的密码、显示留言页每页显示的留言条数等

内容。

以下是该配置文件的全部内容：

```
<?
$file_name="data.dat";                                   //数据文件名
$super_pass="super_man";                                 //管理员名称
$list_num=10;                                            //每页显示留言数
?>
```

把以上代码保存为“18-1.php”，以备后用。

18.4.2　显示提交留言文件建立

这一小节要来创建文本留言板系统的留言显示文件。由于使用了整合功能，所以该文件还包含留言提交前后、留言提交后台等功能。可以说该文件是整个系统中最为重要的一个文件。如果不需要留言修改、留言删除的功能，只要最简单的发布留言、查看留言功能，那么只需要这一个文件就已经够了。可见该文件在整个系统中的重要地位。

一般常见的显示与提交留言在一起的情况是页面上半部分显示留言，下面有一个表单供用户输入留言。这种情况很方便，但影响美观。这里要做的是使用 JavaScript 技术，把用户提交留言的表单放置到一个层中。用户浏览留言时，该层是隐藏的，只有当用户需要留言，并按下相应链接时，才出现该层。关于层的隐藏与显示属于 JavaScript 的内容，不在本书讨论范围，有兴趣的读者，可以查找相关书籍。这里只提供相关功能并不做具体说明。

下面开始具体创建该文件：

```
<?
require "18-1.php";
if(!$_POST[name])                                  //如果没有数据提交，则显示记录
{
?>
<html>
<head>
<title>文本留言板——记录显示页</title>
<script language=javascript>
function Showhide(id,flag)
{
        if(flag==1)
        {
                id.style.display='';
        }
        if(flag==0)
        {
                id.style.display='none';
        }
}
</script>
</head>
```

```
<body>
<center>
<h1>文本留言板——记录显示页</h1>
<p>
<a href=# onClick="Showhide(huifu,1)">签写留言</a>  <a href=18-3.php>管理入口</a>
<p>
<?
if(!$_GET["page"])                              //如果没有参数 page
$page=1;                                        //则显示第一页内容
else
$page=$_GET["page"];                            //如果带有参数 page，则显示相应页内容
if(!file_exists($file_name))                    //如果是第一次运行（文件不存在）
{
    $fp=fopen($file_name,"w");                  //创建文件
    fclose($fp);
}
$myfile=file($file_name);                       //使用 file()函数把所有信息读入一个数组
if($myfile[0]=="")                              //如果文件为空，即没有任何留言信息
echo "目前记录条数为：0";                        //显示没有记录的信息
else
{
    $temp=explode("||",$myfile[0]);             //读出数组第一条记录到数组
    echo "共有".$temp[0]."条内容";               //读出该数组第一个元素（代表记录总条数）
    echo "    ";
    $p_count=ceil($temp[0]/$list_num);          //计算总页数为记录总条数除以每页显示条数
    echo "分".$p_count."页显示";                 //输入总页数
    echo "    ";
    echo "当前显示第".$page."页";                //当前页
    if($page!=ceil($temp[0]/$list_num))         //如果当前页不是最后一页
    $current_size=$list_num;                    //当前页最多可显示$list_num 条记录
    else                                        //如果当前页是最后一页
    $current_size=$temp[0]%$list_num;           //当前页显示的条数为总条数除以$lsit_num 的余数
    if($current_size==0)
    $current_size=$list_num;                    //如果正好是显示条数的倍数则显示$list_num 条内容
    echo "<table border='1'>";
    for($i=0;$i<$current_size;$i++)
    {
        $temp=explode("||",$myfile[($page-1)*$list_num+$i]);//把相应的记录按“||”分割到数组
    echo "<tr>";
        echo "<td>";
        echo "第".$temp[0]."条留言";             //显示记录号
        echo "</td>";
        echo "<td>";
        echo "作者：".$temp[1];                  //显示作者
        echo "</td>";
        echo "<td>";
        echo "写于".$temp[4];                    //显示留言时间
        if($_COOKIE[pass]==$super_pass)         //如果管理者登录，则显示操作连接
        {
```

```
                echo "<a href='18-4.php?id=".$temp[0]."'>修改</a>";
                echo "  ";
                echo "<a href='18-5.php?id=".$temp[0]."'>删除</a>";
            }
            echo "</td>";
            echo "</tr>";
            echo "<tr>";
            echo "<td colspan='3'>";
            echo "主题：".$temp[2];                    //显示留言主题
            echo "</td>";
            echo "</tr>";
            echo "<tr>";
            echo "<td colspan='3'>";
            echo "内容：<br>".$temp[3];                //显示留言内容
            echo "</td>";
            echo "</tr>";
        }
        echo "</table>";
}
echo "<p>";
?>
<div id="huifu" style="display:none; position:absolute; left:15px; top:265px;">
<table border=1 bgcolor="#ffffff">
<form method="post" action=<?echo $PATH_INFO?>>
<tr>
<td colspan="2">请输入留言内容</td>
</tr>
<tr>
<td>姓名：</td>
<td><input name="name" type="text"></td>
</tr>
<tr>
<td>主题：</td>
<td><input name="subject" type="text"></td>
</tr>
<tr>
<td colspan="2">内容：</td>
</tr>
<tr>
<td colspan="2">
<textarea name="content" cols="36" rows="8"></textarea>
</td>
</tr>
<tr>
<td colspan="2">
<input type="submit" name="Submit" value="确定">
<input type="reset" name="Submit2" value="清除">
<input name="Submit3" type="button" onClick="Showhide(huifu,0)" value="关闭">
</td>
```

```
</tr>
</form>
</table>
</div>
<?
//以下内容为分页显示连接
$prev_page=$page-1;                              //前一页
$next_page=$page+1;                              //下一页
if ($page<=1)
{
    echo "第一页 | ";
}
else
{
    echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
}
if ($prev_page<1)
{
    echo "上一页 | ";
}
else
{
    echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
}
if ($next_page>$p_count)
{
    echo "下一页 | ";
}
else
{
    echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
}
if ($page>=$p_count)
{
    echo "最后一页</p>\n";
}
else
{
    echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
}
}
else
{
    $name=$_POST[name];                          //获取作者
    $subject=$_POST[subject];                    //获取主题
    $content=$_POST[content];                    //获取内容
    $time= date(Y 年 m 月 d 日);                   //获取日期
    $s=$name."||".$subject."||".$content."||".$time."\n";  //把内容赋值给变量
    $myfile=file($file_name);                    //使用 file()把记录文件按行读入数组
```

```
		if($myfile[0]=="")                          //如果文件为空
		{
			$fp=fopen($file_name,"a+");             //写入方式打开文件
			fwrite($fp,"1||".$s);                   //直接写入行号为 1 的内容
			fclose($fp);                            //关闭文件
		}
		else
		{
			$temp=explode("||",$myfile[0]);         //把第一条记录按“||”分割到数组
			$temp[0]++;                             //得出总记录数并自增 1
			$fp=fopen($file_name,"r");              //以只读方式打开文件
			$line_has=fread($fp,filesize("$file_name"));  //使用 fread 读出文件已经存在的内容
			fclose($fp);                            //关闭文件
			$fp=fopen($file_name,"w");              //以写入方式打开文件
			fwrite($fp,$temp[0]."||".$s);           //写入新的内容
			fwrite($fp,"$line_has");                //写入原来已经存在的内容
			fclose($fp);                            //关闭文件
		}
	?>
	<html>
	<head>
	<title>文本留言板——记录添加页</title>
	</head>
	<meta http-equiv="refresh" content="2; url=18-2.php">
	<body>
	已经成功更改记录，两秒后返回。
	</body>
	</html>
	<?
	}
	?>
	</body>
	</html>
```

把以上代码保存为“18-2.php”，以备后用。

通过查看以上代码，可以发现主要是使用了对文件的操作，包括按行读取文件内容，把文件内容读入到数组等。下面继续创建其他文件。

18.4.3　管理入口页的创建

管理入口页的作用是把有管理权限的登录资料保存为 Cookie，以便管理者对留言信息做相应的修改。登录密码从配置文件 15-1.php 中获取。

下面就来创建该文件。具体内容参看以下代码：

```
<?
if(!$_POST[pass])                          //如果没有输入管理员名称，则显示 HTML 内容
{
```

```
?>
<html>
<head>
<title>管理入口</title>
</head>
<body topmargin="50" >
<script language=javascript>
function juge(form)
{
	if (form.pass.value == "")
	{
		alert("请输入密码!");
		theForm.pass.focus();
		return (false);
	}
}
</script>
<center>
<h1>管理留言板入口</h1>
<p>
<a href="18-2.php">返回首页</a>
<table border="1">
<form action=<?echo $PATH_INFO?> method=post onsubmit="return juge(this)">
<tr>
<td colspan="2"><center>管理留言板</center></td>
</tr>
<tr>
<td>密码</td>
<td><input type=password name=pass></td>
</tr>
<tr>
<td colspan="2"><center><input type=submit value="登录"></center></td>
</tr>
</form>
</table>
</center>
</body>
</html>
<?
}
else                                                    //如果存在密码，则执行操作
{
	require "18-1.php";                                 //调用配置文件
	if($_POST[pass]!=$super_pass)                       //如果密码错误
	{
?>
<html>
<head>
<title>文本留言板——管理入口</title>
```

```
</head>
<meta http-equiv="refresh" content="2; url=18-3.php">
<body>
登录失败，密码错误！<p>
两秒后返回。
</body>
</html>
<?
}
else                                        //如果密码正确
{
    setcookie("pass","$_POST[pass]");       //注册 Cookie
?>
<html>
<head>
<title>文本留言板——管理入口</title>
</head>
<meta http-equiv="refresh" content="2; url=18-2.php">
<body>
登录成功！<p>
两秒后进入管理页。
</body>
</html>
<?
}
}
?>
```

把以上代码保存为“18-3.php”，以备后用。

该文件没有什么技术含量，只是进行了简单的判断。并根据不同的结果跳转到不到的页面。如果密码正确，则把信息写入 Cookie 数据。

18.4.4　修改留言页面的创建

如果留言中出现了不合适的内容，应修改它们。这时就需要修改留言页面出场了。修改留言的实质是读出某一条留言，读出其中的内容，用新内容代替原有内容，其实质还是对文件的操作。

下面就来创建修改留言页面：

```
<?
require "18-1.php";
if(!$_POST[content])                        //如果没有提交 ID
{
    if(!$_GET[id])                          //如果没有指定修改 ID
    {
        echo "没有指定 ID";
        exit();
    }
```

```
        $id=$_GET["id"];                              //把参数 ID 赋值给变量
        $myfile=file($file_name);                     //使用 file()函数把文件按行读入到数组
        $z=$myfile[0];                                //数组第一个元素赋值给变量
        $temp=explode("||",$myfile[$z-$id]);          //使用 explode 分割相应记录到数组
?>
<html>
<head>
<title>文本留言板——修改留言</title>
</head>
<body>
<center>
<h1>文本留言板——修改留言</h1>
<p>
<h3>只有留言主题与留言内容是可以改变的</h3>
<p>
<a href=18-2.php>返回首页</a>
<table border=1>
<form method="post" action=<?echo $PATH_INFO?>>
<input type=hidden value=<?echo $temp[0]?> name=id>
<tr>
<td colspan="2">以下为第<?echo $temp[0]?>条留言的内容</td>
</tr>
<tr>
<td>姓名：</td>
<td><?echo $temp[1]?></td>
</tr>
<tr>
<td>主题：</td>
<td><input name="subject" type="text" value=<?echo $temp[2]?>></td>
</tr>
<tr>
<td colspan="2">内容：</td>
</tr>
<tr>
<td colspan="2">
<textarea name="content" cols="36" rows="8"><?echo $temp[3]?></textarea>
</td>
</tr>
<tr>
<td colspan="2">
<input type="submit" name="Submit" value="确定">
<input type="reset" name="Submit2" value="清除">
</td>
</tr>
</form>
</table>
</center>
</body>
</html>
```

```
<?
}
else
{
    $id=$_POST[id];                                 //定义 ID
    $subject=$_POST[subject];
    $content=$_POST[content];                       //定义内容
    $myfile=file($file_name);                       //使用 file()函数把文件按行读入到数组
    $z=$myfile[0];                                  //数组第一个元素赋值给变量
    $temp=explode("||",$myfile[$z-$id]);            //使用 explode 分割相应记录到数组
    $s=$temp[0]."||".$temp[1]."||".$subject."||".$content."||".$temp[4];
    for($i=0;$i<($z-$id);$i++)
    {
        $text2=$text2.$myfile[$i];                  //内容保持不变
    }
    for($i=($z-$id+1);$i<$z;$i++)                   //新欲删除的后一条记录到最后一条记录
    {
        $text1=$text1.$myfile[$i];                  //内容保持不变
    }
    $fp=fopen($file_name,"w");                      //以写入方式打开文件（文件同时被清空）
    fwrite($fp,$text2);                             //写入欲删除记录之前的所有记录
    fwrite($fp,$s);                                 //写入更改过的记录
    fwrite($fp,$text1);                             //写入欲删除记录之后的所有记录
    fclose($fp);                                    //关闭文件
?>
<html>
<head>
<title>文本留言板——修改留言</title>
</head>
<meta http-equiv="refresh" content="2; url=18-2.php">
<body>
修改成功！<p>
两秒后返回。
</body>
</html>
<?
}
?>
```

将以上代码保存为“18-4.php”，以备后用。

以上代码，实现了对文本文档的复杂操作。不过从以上代码可以发现，文本留言板相当低效。为了修改一条留言，就不得不把所有留言内容读入内存。不过好在现在连最一般的 PC 机处理数据的速度也是相当快的，所以使用文本留言板的速度瓶颈并不是特别明显。

18.4.5　删除留言页面的创建

网上的所有留言板都面临着一个问题的困扰，那就是“恶意灌水”。对那些恶意灌水的留言记录，

要及时删除。这样一方面可以减少空间的占用，另外还可以使整个留言本页面显得干净整洁。

这一小节就来创建删除留言页面。具体内容参考以下代码：

```
<?
require "18-1.php";
if(!$_GET[id])                              //如果没有指定删除 ID
{
    echo "没有指定 ID";
    exit();
}
$id=$_GET["id"];                            //把参数 ID 赋值给变量
$myfile=file($file_name);                   //使用 file()函数把文件按行读入到数组
$z=$myfile[0];                              //数组第一个元素赋值给变量
$temp=explode("||",$myfile[$z-$id]);        //使用 explode 分割相应记录到数组
for($i=0;$i<($z-$id);$i++)
{
    $temp2=explode("||",$myfile[$i]);       //使用 explode 分割相应记录到数组
    $temp2[0]--;                            //记录号实现自减
    $text2=$text2.$temp2[0]."||".$temp2[1]."||".$temp2[2]."||".$temp2[3]."||".$temp2[4];
                                            //把新的内容赋值到变量
}
for($i=($z-$id+1);$i<$z;$i++)               //新欲删除的后一条记录到最后一条记录
{
    $text1=$text1.$myfile[$i];              //内容保持不变
}
$fp=fopen($file_name,"w");                  //以写入方式打开文件（文件同时被清空）
fwrite($fp,$text2);                         //写入欲删除记录之前的所有记录
fwrite($fp,$text1);                         //写入欲删除记录之后的所有记录
fclose($fp);                                //关闭文件
?>
<html>
<head>
<title>文本留言板——删除留言</title>
</head>
<meta http-equiv="refresh" content="2; url=18-2.php">
<body>
删除成功！<p>
两秒后返回。
</body>
</html>
```

把以上代码保存为“18-5.php”，以备后用。

该文件的实现方法是先读出欲删除留言记录之前的所有记录，并把它们的记录号全部自减 1。再读出欲删除留言记录后的所有记录，保持不变。然后，再把修改过的记录全部写入文件，这样原有的欲删除的记录就会被从文件中“删除”，实质上是重新写入了不被删除的文件。

至此，该文本留言板的所有页面就全部完成了。18.5 节将对该系统做一下测试，测试该文本留言板各项功能是否都能正常运行。

18.5　调试运行留言板

18.4 节，已经把留言板所需的模块全部创建完毕，这一节就对留言板做全面测试。

（1）先在 PHP 执行环境下运行 18-2.php。由于是第 1 次运行，所以结果会如图 18.1 所示。由于此时还没有任何留言，所以显示记录条数为 0。到 18-2.php 所在的路径下查看会发现多了一个名为“data.dat”的空文件。说明创建数据文件成功。单击“签写留言”链接来测试插入记录功能是否正常。单击链接后，在页面下方会出现输入表单，如图 18.2 所示。

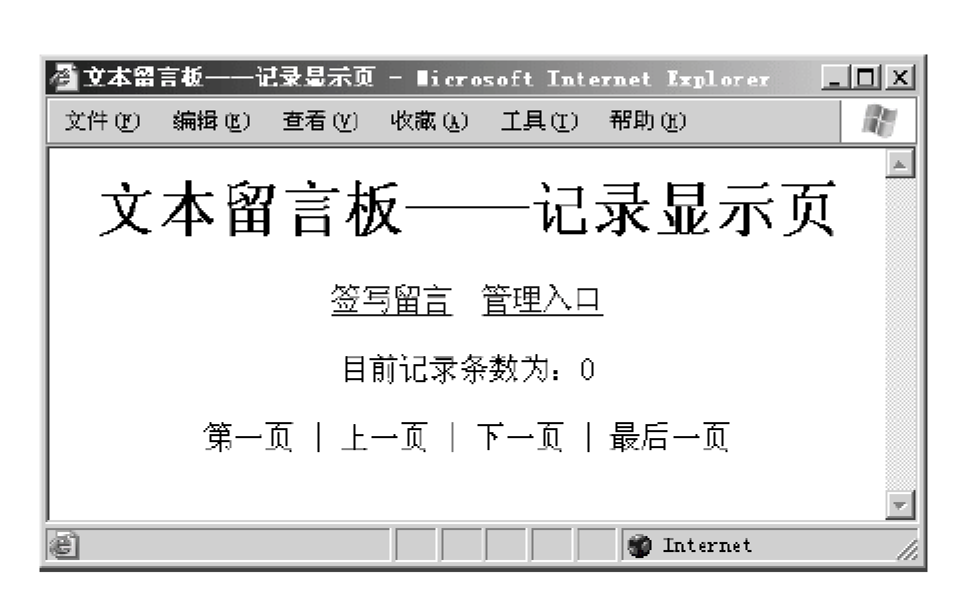


图 18.1　第 1 次执行显示留言页执行结果



图 18.2　显示隐藏的表单效果

（2）在如图 18.2 所示的表单中输入相应的内容，然后单击“确定”按钮，会出现如图 18.3 所示的结果。

（3）留言添加成功后，直接跳转回来。从图 18.3 可以看出，留言已经成功添加并显示，打开留言板程序目录下的 data.dat 文件，其中内容如下所示。

1||无名氏||今天天气不错||今天天气不错，不太冷也不太热。||2006 年 10 月 05 日

可见，已经将相关信息正确添加至记录文件了。

（4）下面来测试管理入口能不能进。单击“管理入口”链接后，出现如图 18.4 所示的画面。

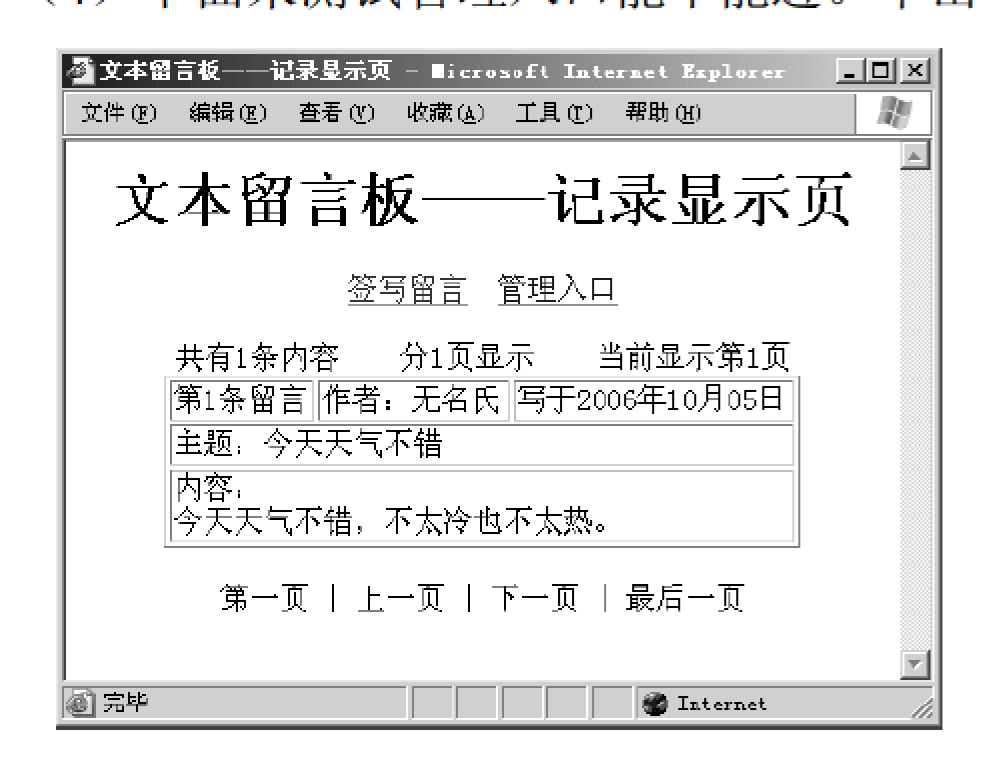


图 18.3　添加留言执行结果



图 18.4　管理留言板执行结果

（5）输入正确的密码（这里输入前面 18-1.php 中定义的$super_pass 的值），并单击“登录”按钮后，结果会如图 18.5 所示。对比图 18.3 与图 18.5 的区别，可以发现，图 18.5 比图 18.3 上每一条留言后面都多了修改与删除的链接。说明已经正常登录并可以进行相关操作。

（6）由于现在留言板中记录较少，所以继续添加一些记录，以测试相关内容。单击图 18.5 中的修改链接，将打开如图 18.6 所示的画面。

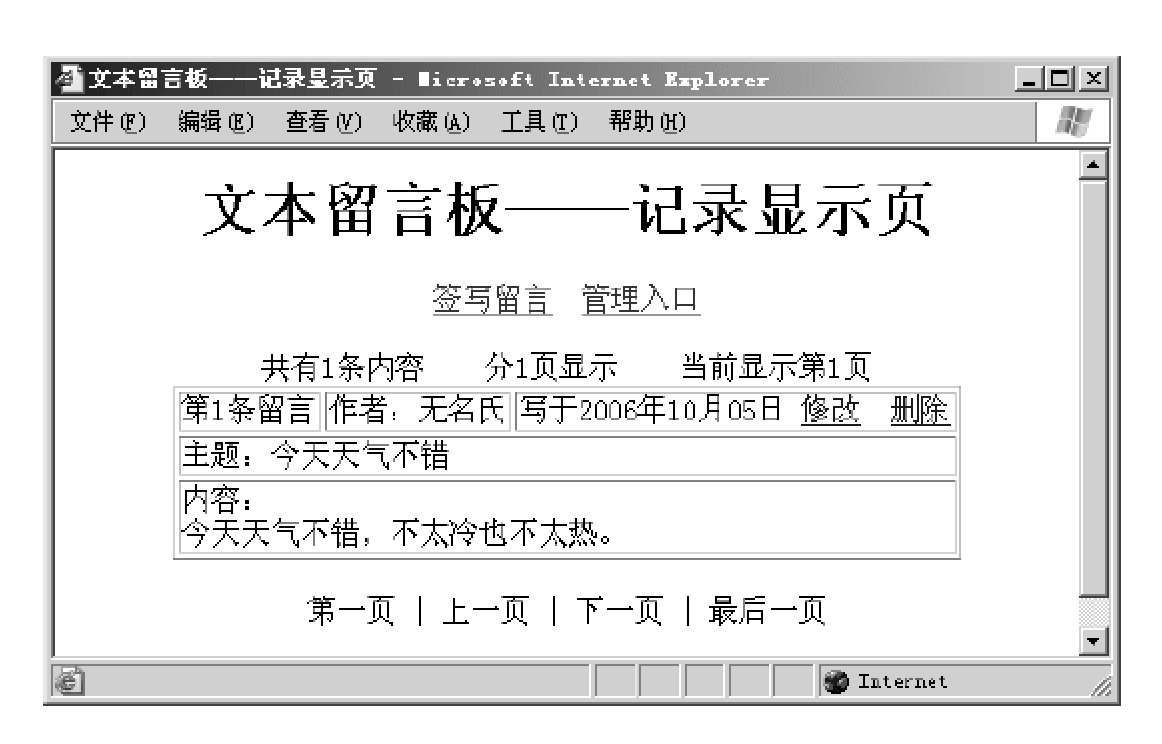


图 18.5　正确登录管理执行结果

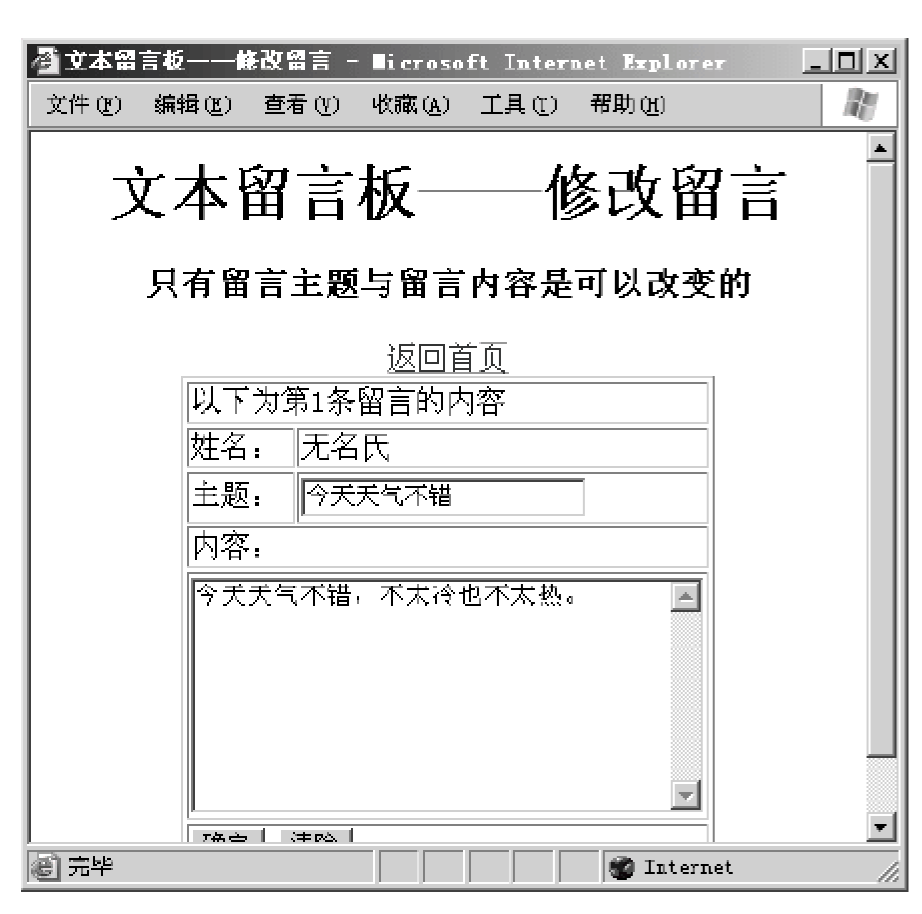


图 18.6　修改记录的前台

（7）按照需要进行修改，这里把“今天天气不错”，改为“今天天气为什么这么不错？”，其他的内容保持不变，单击下方的“确定”按钮。执行结果如图 18.7 所示。从图 18.7 的执行结果可以看出，留言内容已经按照用户的输入进行了更改，说明修改留言功能运行正常。

（8）下面测试程序的删除功能。删除前的记录情况如图 18.8 所示。

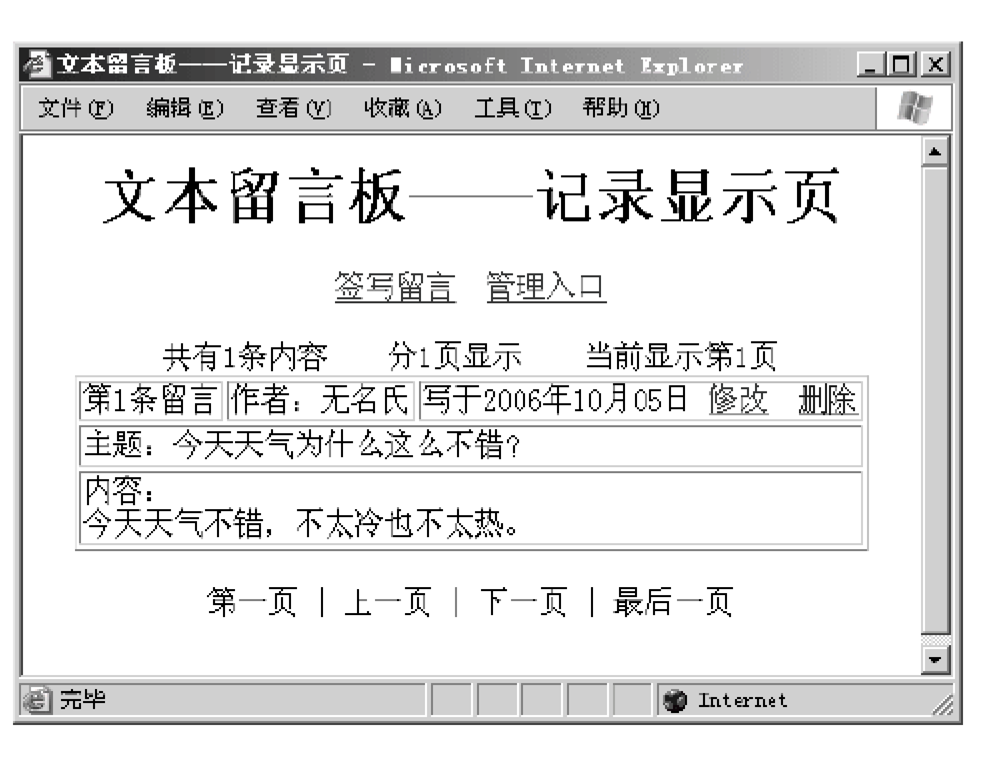


图 18.7　修改后的结果

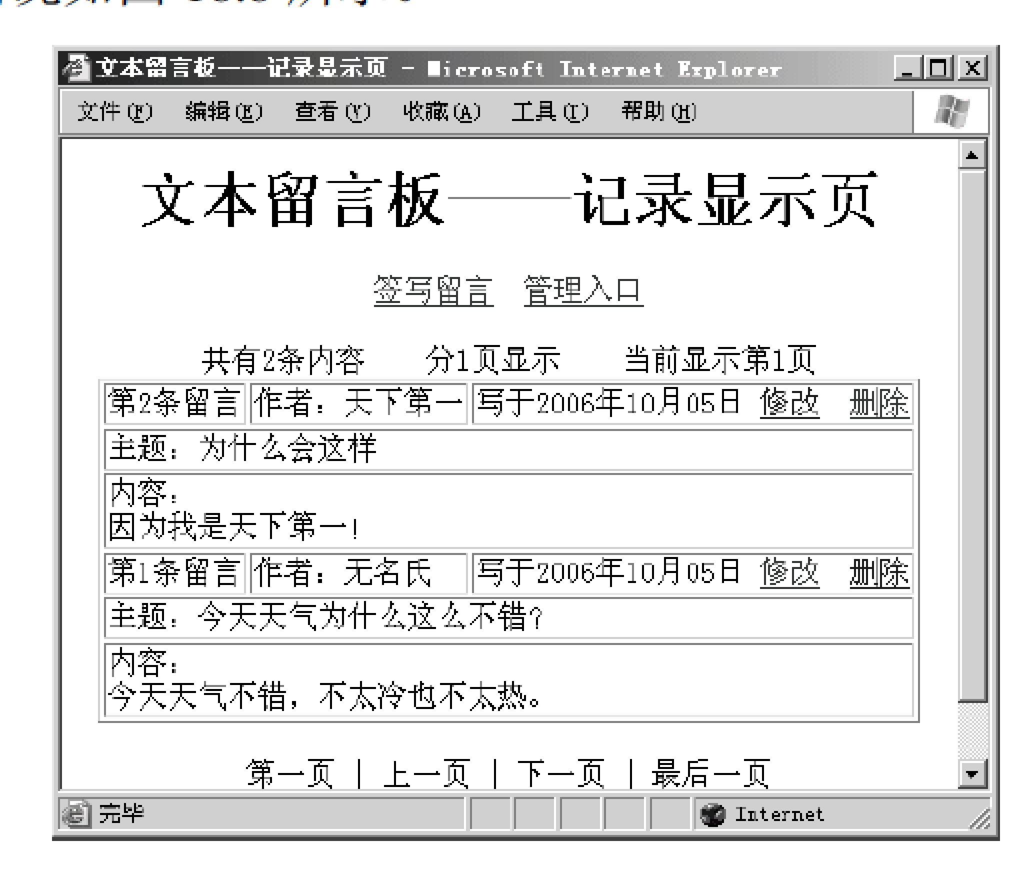


图 18.8　删除操作前的效果

（9）下面把图 18.8 中的第 2 条留言删除。单击“删除”链接执行 18-5.php。执行后的结果如图 18.9 所示。

（10）对比图 18.8 与图 18.9 可以发现：留言数量由原来的 2 条变成了 1 条，有一条信息已经被删除。至此整个系统全部测试完毕。

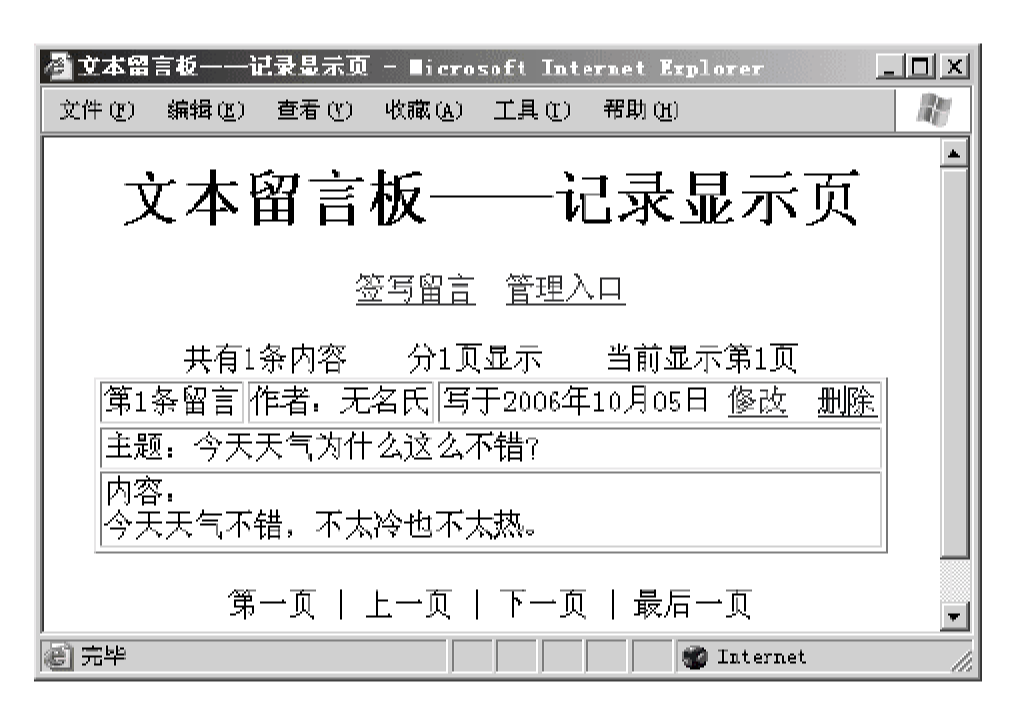


图 18.9　删除操作后的结果

18.6　进一步完善

这个文本留言板对于个人用户来说基本上已经够使用了。不过，如果对要求比较多的用户来说，该留言板还是有需要完善的地方。其一，留言板没有回复功能。其二，留言板存在安全隐患。其三，留言板的内容还不够丰富，如提交留言时没有可以使用表情图片之类的功能。

对于这些问题，作者只提供解决思路，有兴趣的读者，可以自行研究以丰富这个文本留言板。

对于第 1 个问题，留言板没有回复功能，解决起来相对麻烦一些，可以给留言数据在原来的 5 项，留言 ID、作者、题目、内容、留言时间上再增加一项 r_id。该项只记录父留言的 ID，如果某一条留言本身就是父留言，则 r_id 的值为 0。然后在显示一条编号为 ID 的留言时，再遍历它后面所有 r_id=父留言 ID 的记录作为回复显示。这样就可以了，虽然这样能够实现，但在删除留言特别是父留言时就会相当麻烦。这也是文本留言板自身的一个局限性。

现在这个留言板有什么安全隐患呢？因为没有对用户的输入进行相应的屏蔽，所以用户可以使用 HTML 代码。这在通常情况下是不被允许的。不过解决起来很简单，就是使用 htmlspecialchars()函数对用户的输入做相应的处理。这样输入的 HTML 代码就会被变成普通的文本，不再具有威胁性了。

添加表情，也不复杂，只需给原来的数据上再增加一项，里面存放小图片的链接。显示时，显示相应的表情符号或者小图片即可。

18.7　本 章 小 结

至此，关于文本留言板的内容就全部为读者介绍完了。虽然文本留言板有其局限性，功能上还不够强大，但只要善于通过不同的方法去解决现实问题，文本留言板还是有用武之地的。通过本章的学习，相信读者对于如何建立并使用一个文本留言板都应该有一个深刻的认识了。说到底，文本留言板其实质还是对文本文件的操作。这一章就为读者介绍到这里。

第19章

PHP 博客程序

- 什么是博客
- 博客程序的实现原理
- 安装文件制作
- 日志显示
- 用户注册、登录
- 留言功能
- 类比添加

如果要问当前网络上最流行的网络应用是什么，真正的答案是博客。关于博客的说法和比喻，网上有这样几种：1、博客是继 E-mail、BBS、ICQ 之后出现的第 4 种网络交流方式。2、博客是网络时代的个人“读者文摘”。3、博客是以超级链接为武器的网络日记。4、博客是信息时代的麦哲伦。5、博客代表着新的生活方式和新的工作方式，更代表着新的学习方式。通过博客，让自己学到很多，让别人学到更多。6、博客代表着“新闻媒体 3.0 版”，旧媒体（Old Media）→新媒体（New Media）→自媒体（We Media）。

总之，博客是一个正处于快速发展和快速演变中的互联网新应用。所以，对于一个 PHPer（PHP 程序员）来说，了解如何写出高效的博客程序是十分有必要的。这一章就向读者介绍一下如何用 PHP 和 MySQL 数据库结合来制作一个简单的博客程序。本章内容包括什么是博客、简单的博客程序的功能、博客程序的代码实现等。通过本章的学习，会使读者对什么是博客、如何写出博客程序有本质上的认识。

19.1　什么是博客

要想写出博客程序，第 1 步就是要先搞明白，到底什么是博客，这一节就来回答这个问题。

中文“博客”一词，源于英文单词 Blog/Blogger。Blog 是 Weblog 的简称。Weblog 其实是 We 和 Log 的组合词。Web 指 World Wide Web，当然是指互联网了；Log 的原义则是“航海日志”，后指任何类型的流水记录。合在一起理解，Weblog 就是在网络上的一种流水记录形式或者简称“网络日志”。

Blogger 或 Weblogger 是指习惯于日常记录并使用 Weblog 工具的人。虽然在大陆早些时候或者在我国台湾等地，对此概念的译名不尽相同（有的称为“网志”，有的称之为“网录”等），但目前已基本统一到“博客”一词上来。该词最早是在 2002 年 8 月 8 日由著名的网络评论家王俊秀和方兴东共同撰文提出来的。博客也好，网志也罢，仅仅是一种名称而已，它的本义还是逃不过 Weblog 的范围。只是，通常所说的“博客”，既可用作为名词 Blogger 或 weblogger——指具有博客行为的一类人；也可以作动词用（相当于英文中的 Weblog 或 blog），指博客采取的具有博客行为反映、是第三方可以用视觉感受到的行为，即博客们所撰写的 Blog。因此，“他/她是一位博客，他/她天天在博客”及“博客博什么客?”在中文语法与逻辑上都是正确的。只是不同场合的用法不同罢了。

Blog 究竟是什么？说了半天，其实一个 Blog 就是一个网页，它通常是由简短且经常更新的帖子（Post）所构成，这些张贴的文章都按照年份和日期倒序排列。Blog 的内容和目的有很大的不同，从对其他网站的链接、评论，有关公司、个人构想的新闻，到日记、照片、诗歌、散文，甚至科幻小说的发表或张贴都有。许多 Blogs 只是记录着 blog 个人所见、所闻、所想，还有一些 Blogs 则是一群人基于某个特定主题或共同利益领域的集体创作。撰写这些 Weblog 或 Blog 的人就叫做 Blogger 或 Blog writer。

19.2　简单博客程序的功能

19.1 节为读者介绍了什么是博客。这一节，就接着来讲，简单博客程序所应具有的功能。

通常一个最简单的博客程序应具有以下功能：

- 博客所有者可以添加日志。
- 普通浏览者无权添加日志，只能对已经存在的内容进行评论。
- 所有者可以管理日志。内容包括删除、编辑日志；增加、删除更改日志的类别等，以及可以设定日志的保密状态，注册用户与普能用户的权限等。
- 通常情况下，日志还附带一个留言板，供普通浏览者与博客的所有者进行有效的交流。
- 可以用一个简单的用户注册、登录系统，以使普通浏览者与注册用户有不同的权限。

19.3　制作前的准备工作

通过 19.1 节和 19.2 节的介绍，读者对于什么是博客程序，一个简单的博客程序所应具备的功能应

该有一个大致的认识了。从这一节开始，就来通过 PHP 代码来一步步实现简单的博客程序的所有功能。

19.3.1　配置文件的创建

配置文件是程序各种资源访问的源头。其中存放着连接博客程序数据库的主机名、连接主机的用户名、密码、数据库名、表名等重要信息。

具体内容参看以下代码：

```
<?
$host="localhost";                          //主机名
$user="root";                               //用户名
$pass="";                                   //密码
$db_name="test";                            //数据库名
$table_log="b_log";                         //日志表
$table_user="b_user";                       //用户表
$table_gbook="b_gbook";                     //留言表
$table_sort="b_sort";                       //类别表
?>
```

把以上代码保存为“19-1.php”， 以备后用。

19.3.2　安装文件的创建

这一小节，来创建一个安装文件。该文件的作用是从配置文件（19-1.php）中获取基本信息，然后提供给用户一个表单。当用户按要求填写所需数据后，执行创建表及插入记录的操作，完成程序表的创建。

下面给出具体的代码：

```
<?
if(!$_POST[admin])                          //如果没有默认参数，则显示 HTML
{
    echo "<html>\n";
    echo "<head>\n";
    echo "<title>安装程序</title>\n";
    echo "</head>\n";
    echo "<body>\n";
    echo "<script language=\"javascript\">\n";
    echo "function juge(theForm)\n";
    echo "{\n";
    echo "\tif (theForm.admin.value == \"\")\n";
    echo "\t{\n";
    echo "\t\talert(\"请输入管理员名称！ \");\n";
    echo "\t\ttheForm.admin.focus();\n";
    echo "\t\treturn (false);\n";
    echo "\t}\n";
    echo "\tif (theForm.pass.value == \"\")\n";
```

```
echo "\t{\n";
echo "\t\talert(\"请输入管理员密码！\");\n";
echo "\t\ttheForm.pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.pass.value.length < 8 )\n";
echo "\t{\n";
echo "\t\talert(\"密码至少要 8 位！\");\n";
echo "\t\ttheForm.pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.re_pass.value !=theForm.pass.value)\n";
echo "\t{\n";
echo "\t\talert(\"确认密码与密码不一致！\");\n";
echo "\t\ttheForm.re_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.nickname.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入昵称！\");\n";
echo "\t\ttheForm.nickname.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">\n";
echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">安装博客</font></td>\n";
echo "</tr>\n";
echo "<tr   bgcolor=\"#cccc99\">\n";
echo "<td>管理员：（后台登录）</td>\n";
echo "<td><input type=\"text\" name=\"admin\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>管理员密码：（不小于 8 位）</td>\n";
echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>确认密码：</td>\n";
echo "<td><input type=\"password\" name=\"re_pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>管理员 E-mail：（可选）</td>\n";
echo "<td><input type=\"text\" name=\"email\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>管理员昵称：（前台显示）</td>\n";
```

```
        echo "<td><input type=\"text\" name=\"nickname\"></td>\n";
        echo "</tr>\n";
        echo "<tr bgcolor=\"#cccc99\">\n";
        echo "<td colspan=\"2\"><center><input type=\"submit\" value=\"下一步\"></center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
}
else                                            //如果有 POST 参数执行操作
{
        $username=$_POST[admin];                //获得参数
        $password=md5($_POST[pass]);
        $nickname=$_POST[nickname];
        $email=$_POST[email];
        require "19-1.php";
        $link=mysql_connect("$host","$user","$pass") or die(mysql_error());
        mysql_select_db($db_name,$link);        //选择数据库
        $sql="create table $table_log(
        id int(5) not null auto_increment primary key,
        p_id int(5) not null default 0,
        title varchar(40) not null default '',
        content text not null default '',
        sort varchar(20) not null default '',
        views int(5) not null default 0,
        tbcount int(5) not null default 0,
        author varchar(40) not null default '',
        date varchar(20) not null default '',
        top enum('n','y') not null default 'n',
        hide enum('n','y') not null default 'n',
        allow_tb enum('n','y') not null default 'n'
        )";
        mysql_query($sql,$link) or die(mysql_error());   //发送创建 B_LOG 表的 SQL 请求
        $sql="create table $table_user(
        id int(5) not null auto_increment primary key,
        username varchar(40) not null default '',
        password varchar(40) not null default '',
        admin enum('1','0') not null default '0',
        nickname varchar(20) not null default '',
        sex enum('boy','girl') not null default 'boy',
        photo varchar(80) not null default '',
        email varchar(60) not null default '',
        description varchar(200) not null default ''
        )";
        mysql_query($sql,$link) or die(mysql_error());   //发送创建 B_USER 表的 SQL 请求
        $sql="create table $table_gbook(
        id int(5) not null auto_increment primary key,
```

```
        title varchar(40) not null default '',
        content text not null default '',
        author varchar(40) not null default '',
        date varchar(30) not null default ''
        )";
        mysql_query($sql,$link) or die(mysql_error());  //发送创建 B_GBOOK 表的 SQL 请求
        $sql="create table $table_sort(
        id int(5) not null auto_increment primary key,
        sortname varchar(20) not null default '',
        sortimg varchar(60) not null default '',
        sortnum int(5) not null default 0,
        description varchar(200) not null default ''
        )";
        mysql_query($sql,$link) or die(mysql_error());  //发送创建 B_SORT 表的 SQL 请求
        $sql="insert into $table_sort(sortname,sortimg,description)values('默认类别','images\sort.img','默认的分
    类，请更改！')";
        mysql_query($sql,$link) or die(mysql_error());  //发送添加默认分类的 SQL 请求
        $sql="insert                                                                                 into
    $table_user(username,password,admin,nickname,email,description)values('$username','$password','1','$nickn
    ame','$email','我就是本小站的管理员！')";
        mysql_query($sql,$link) or die(mysql_error());  //发送添加管理员信息的 SQL 请求
        echo "<html>\n";
        echo "<head>\n";
        echo "<title>安装程序</title>\n";
        echo "</head>\n";
        echo "<body>\n";
        echo "</center>";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">\n";
        echo "<tr bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"5px\">安装博客</font></td>\n";
        echo "</tr>\n";
        echo "<tr   bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"3px\">成功安装！</font></td>\n";
        echo "</tr>\n";
        echo "<tr   bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\">点<a href=\"19-5.php\">这里</a>进入</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        echo "</center>";
        echo "</body>\n";
        echo "</html>\n";
    }
    ?>
```

把以上代码保存为“19-2.php”，然后在 PHP 执行环境下运行该 PHP 文件，以完成各种所需要的表的创建。其首次执行结果如图 19.1 所示。按图 19.1 的要求填入用户信息，然后单击“下一步”按钮，将会开始各种表的创建工作。进度完成之后的结果如图 19.2 所示。

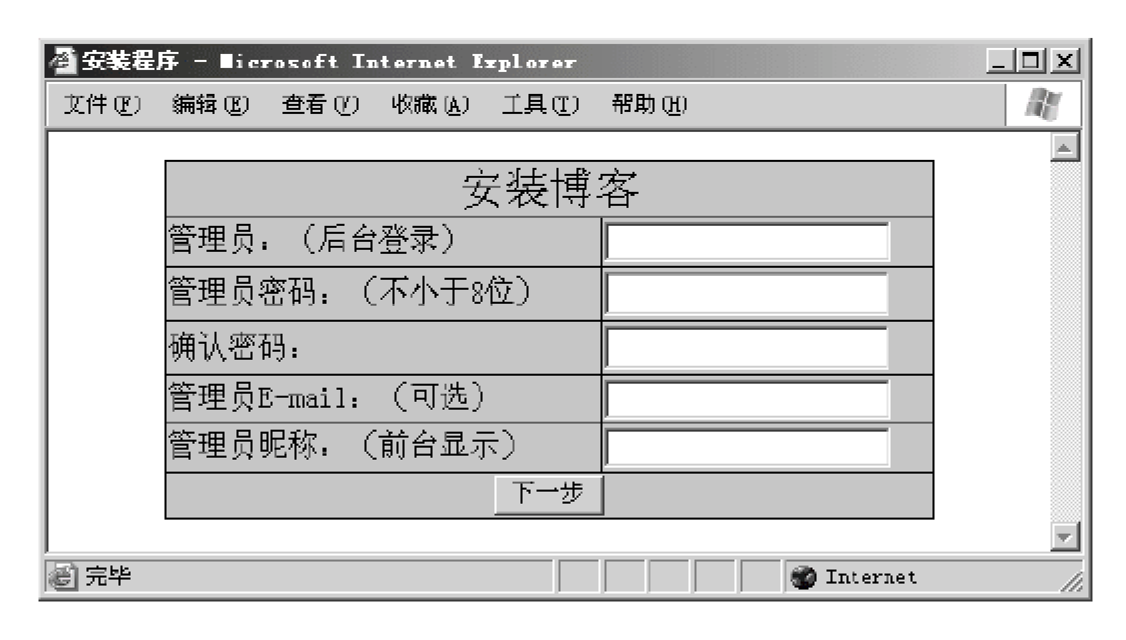


图 19.1　安装文件首次运行的结果



图 19.2　安装完成后的执行结果

这时各种表已经创建完成了。可以登录进去体验一下博客的魅力了，但是现在还不能单击“点这里进入”链接，因为相关的页面还没有被创建。下一步，就来创建登录的页面。

19.3.3　头文件的创建

头文件，显示页面的头部分。单独做成一个文件，便于在多处调用，以减少代码的体积。具体内容参看以下代码：

```
<?
echo "<html>\n";
echo "<head>\n";
echo "<title>博客程序</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<style type=\"text/css\">";
echo "p {   font-size: 12px; color:#0000ff}";
echo "table { table-layout: fixed; word-wrap:break-word;}";
echo "td {text-align: left; text-valign:center; font-size: 12px;}";
echo "body {   font-size: 12px;   text-decoration: none;color:#3300cc}";
echo "a {   font-size: 12px; text-decoration: none; color: #0000FF}";
echo "a:hover { color: #FF0000; text-decoration: underline}";
echo "a:link {   text-decoration: none}";
echo "</style>";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\"     bgcolor=\"#000000\">\n";
echo "<tr>\n";
echo "<td bgcolor=\"#ffffff\"></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"19-5.php\">首页</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"19-6.php\">日志</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"19-17.php\">留言</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"19-8.php\">管理</a></center></td>\n";
echo "</tr>\n";
echo "</table>\n";
echo "<p>";
?>
```

把以上代码保存为“19-3.php”备用，以便其他页面能够调用该文件，以减少代码量。

19.3.4　侧边文件的创建

通常的博客程序侧边显示的内容也是基本相同的，与头文件一样，侧边也可以单独做成一个文件，需要时调用即可。这样做的好处显而易见，如减少代码量，减轻程序更改工作量。如果需要对某一部分进行修改，直接修改该文件即可，而不用一个页面文件一个页面文件的修改。侧边显示以下内容：

- 博客管理员信息。
- 用户登录表单。
- 如果用户已经登录则显示登录用户名称。
- 日历，以便使用者可以根据不同日期查看当天的日志。
- 日志的所有类别。方便使用者选择不同的类别。

具体内容参看以下代码：

```
<?
echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">\n";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";
require "19-1.php";                                    //调用配置文件
$link=mysql_connect($host,$user,$pass);                //连接主机
mysql_select_db($db_name,$link);                       //选择数据库
$sql="select * from $table_user where admin='1'";      //显示管理员信息
$result=mysql_query($sql,$link);                       //发送 SQL 请求
$row=mysql_fetch_array($result);                       //获取信息
echo "管理员昵称：";
echo $row[nickname];                                   //显示昵称
echo "<br>";
echo "管理员性别：";
echo $row[sex];                                        //显示性别
echo "<br>";
echo "管理员信箱：";
echo $row[email];                                      //显示信箱
echo "<br>";
echo "管理员图像：";
if($row[photo]) echo "<img src=images/".$row[photo].">"; //如果有图像，则显示图像
echo "<br>";
echo "管理员自白：<br>";
echo $row[description];                                //显示自白
echo "</td>";
echo "</tr>";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";
if(!$_COOKIE[username])                                //如果没有登录用户
{
echo "<form action=19-8.php method=post>";             //显示登录表单
echo "用户名：<input type=text name=username size=5><br>";
```

```
echo "密码：<input type=password name=password size=6><br>";
echo "<input type=submit value=\"登录\">";
echo "<input type=button value=\"注册\" onclick=w_open()>";
echo "</form>";
}
else                                                    //如果有登录用户
{
echo "欢迎你:$_COOKIE[username]";                       //显示登录用户名
echo "<p align=\"center\">";
echo "<a href=19-18.php>退出</a>";                      //显示退出连接
}
echo "</td>";
echo "</tr>";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";                        //以下用 JS 显示一个日历
echo "<form name=f>
<select name=year size=1 onchange=change()>";
for($i=2005;$i<2016;$i++)
{
    echo "<option value=".$i;
    if($i==date("Y")) echo " selected ";
    echo ">".$i;

}
echo "</select>
<select name=month size=1 onchange=change()>";
for($i=1;$i<13;$i++)
{
    echo "<option value=".$i;
    if($i==date("n")) echo " selected ";
    echo ">".$i;

}
echo "</select></form>
<div id=a1></div>
<script language=javascript>
tian=new Array();
tian[0]=31;
tian[1]=28;
tian[2]=31;
tian[3]=30;
tian[4]=31;
tian[5]=30;
tian[6]=31;
tian[7]=31;
tian[8]=30;
tian[9]=31;
tian[10]=30;
tian[11]=31;
```

```
function change()
{
        y=document.f.year.value;
        m=document.f.month.value;
        n=0,s=0;
        if(y%4==0) tian[1]=29;
        for(i=2005;i<y;i++)
        {
                if(i%4==0) n=366;
                else n=365;
                s=s+n;
        }
        for(i=1;i<m;i++)
        {
                s=s+tian[i-1];
        }
        w=(s+6)%7;
        a1.innerHTML=\"\";
        content=\"<table><tr><td> 日 </td><td> 一 </td><td> 二 </td><td> 三 </td><td> 四 </td><td> 五 </td><td> 六
</td></tr>\";
        t=1;
        for(i=0;i<42;i++)
        {
                if(i%7==0)
                {
                        content=content+\"<tr>\";
                }
                if((i>=w)&&(i<(tian[m-1])+w))
                {
                        content=content+\"<td><a href=19-6.php?y=\"+y+\"&m=\"+m+\"&d=\"+t+\">\"+t+\"</a></td>\";
                        t++;
                }
                else
                {
                        content=content+\"<td> </td>\";
                }
                if(i%7==6)
                {
                        content=content+\"</tr>\";
                }
        }
        content=content+\"</table>\";
        a1.innerHTML=content;
}
change();
function w_open()
{
open(\"19-18.php\");
}
```

```
</script>";
echo "</td>";
echo "</tr>";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";
echo "日志类别：<br>";
$sql="select sortname,sortnum from $table_sort";           //显示所有日志类别
$result=mysql_query($sql,$link);                           //发送 SQL 请求
while($rows=mysql_fetch_array($result))                    //循环显示类别名及数量
{
    echo "<a href=19-6.php?sort=".$rows[0].">".$rows[0]."</a>（".$rows[1]."）";
    echo "<br>";
}
echo "</td>";
echo "</tr>";
echo "</table>";
?>
```

把以上代码保存为“19-4.php”，以备后用。

19.4　日志显示模块

19.3 节，介绍了在创建程序之前所要做的准备工作。并建立了配置文件，创建了相应的数据库、表，还建立了显示时需要调用的头文件与侧边文件，这一节来实现日志显示模块。该模块共有以下几部分组成，如主显示页面、日志显示页面、日志详细信息页面等。下面分别为读者做介绍。

19.4.1　主显示页面的创建

这一小节来创建该博客程序的主页文件。主页页面即用户首次可以看到的页面，除了调用头文件与侧边文件显示内容之外，它还有以下几个内容，即显示最新日志、显示最新留言。更多具体内容请参看以下代码：

```
<?
require "19-1.php";                                        //调用配置文件
require "19-3.php";                                        //调用头文件
echo "<table width=\"80%\">\n";
echo "<tr>";
echo "<td width=\"20%\">";
require "19-4.php";
echo "</td>";
echo "<td width=\"80%\">";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" align=\"center\" bgcolor=\"#000000\">\n";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">最新日志</td>";
echo "</tr>";
```

```
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";                          //显示最新 10 条日志
$sql="select * from $table_log where p_id=0 order by id desc limit 10";
$result=mysql_query($sql,$link);                          //发送 SQL 请求
$nums=mysql_num_rows($result);                            //获取日志数量
if($nums<1) echo "还没有任何日志记录！";                  //如果没有日志显示信息
else                                                      //如果有日志
{
    while($rows=mysql_fetch_array($result))               //通过循环显示内容
    {
        echo "<a href=19-5.php?id=".$rows[id].">".$rows[title]."</a>";
        echo "<p>";
        echo $rows[date];
        echo "<p>";
        echo $rows[content];
        echo "<p>";
        echo "<a href=19-7.php?id=".$rows[id].">阅读全文</a>|";
        echo "<a href=19-6.php?sort=".$rows[sort].">类别：".$rows[sort]."</a>|";
        echo "<a href=19-7.php?id=".$rows[id].">评论（".$rows[tbcount]."）</a>|";
        echo "<a href=19-7.php?id=".$rows[id].">浏览（".$rows[views]."）</a>";
        echo "<hr width=\"100%\">";
    }
}
echo "</td>";
echo "</tr>";
echo "</table>";
echo "<p>";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" align=\"center\" bgcolor=\"#000000\">\n";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">最新留言</td>";
echo "</tr>";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";                          //显示最新 10 条留言
$sql="select * from $table_gbook order by id desc limit 10";
$result=mysql_query($sql,$link);                          //发送 SQL 请求
$nums=mysql_num_rows($result);                            //获取留言数量
if($nums<1) echo "还没有任何留言！";                      //没有没有留言显示信息
else                                                      //如果有留言
{
    while($rows=mysql_fetch_array($result))               //通过循环显示内容
    {
        echo $rows[author]."于".$rows[date]."留言说：";
        echo "<p>";
        echo $rows[content];
        echo "<hr width=\"100\">";

    }
}
echo "</td>";
```

```
echo "</tr>";
echo "</table>";
echo "</td>";
echo "</tr>";
echo "</table>";
echo "</center>";
echo "</body>";
echo "</html>";
?>
```

把以上代码保存为“19-5.php”。在 PHP 执行环境中运行该文件，其执行结果如图 19.3 所示。为什么会这样？因为这时并没有任何日志，也没有任何留言，所以才会显示如图 19.3 所示的执行结果。如果以管理员身份登录系统并添加相应的类别及日志，首页面执行结果会如图 19.4 所示。



图 19.3　第 1 次运行主显示页面执行结果



图 19.4　第 2 次执行首页显示页面执行结果

此时的首页面正确显示出了相应的日志及日志类别，说明程序正常运行。

19.4.2　日志显示页面的创建

这一小节来创建日志显示页，该页面的作用是显示所有已经存在的日志，显示单条日志的浏览量、评论量等内容，并且还具有分页功能。同时还应该根据用户不同的选择显示不同的内容，如用户选择了某一天的日志，则只显示某一天的内容；如果用户选择了某一类别的日志，则只显示某一类别的内容。

更多具体内容参看以下代码：

```
<?
require "19-3.php";                         //调用头文件
echo "<table width=\"80%\">\n";
echo "<tr>";
echo "<td width=\"20%\">";
require "19-4.php";                         //调用侧边文件
echo "</td>";
echo "<td width=\"80%\">";
```

```
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" align=\"center\" bgcolor=\"#000000\">\n";
require "19-1.php";                              //调用配置文件
$link=mysql_connect($host,$user,$pass);
mysql_select_db($db_name,$link);
if(!($_GET[y]and$_GET[m]and$_GET[d]))            //如果没有日期输入
{
    if($_GET[sort])                              //如果有类别输入
    {
        $sort=$_GET[sort];                       //获取类别
        echo "<tr>";
        echo "<td bgcolor=\"#eeeeff\">分类<font size=\"3\">[".$sort."]</font>的日志</td>";
        echo "</tr>";                            //显示相应类别
        $sql="select * from $table_sort where sortname='$sort'";
        $result=mysql_query($sql,$link);
        $rows=mysql_fetch_array($result);
        echo "<tr>";
        echo "<td bgcolor=\"#eeeeff\">类别介绍：".$rows[description]."</td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeeeff\">本类别共有日志：".$rows[sortnum]."篇</td>";
        echo "</tr>";
        if($rows[sortnum]>0)                     //如果有类别日志
        {
            echo "<tr>";
            echo "<td bgcolor=\"#eeeeff\">";
            $sql2="select * from $table_log where sort='$sort' and p_id=0";
            $result2=mysql_query($sql2,$link);
            while($rows2=mysql_fetch_array($result2))//循环显示
            {
                echo "<a href=19-7.php?id=".$rows2[id].">".$rows2[title]."</a>";
                echo "<p>";
                echo $rows2[date];
                echo "<p>";
                echo $rows2[content];
                echo "<p>";
                echo "<a href=19-7.php?id=".$rows2[id].">阅读全文</a>|";
                echo "<a href=19-6.php?sort=".$rows2[sort].">类别".$rows2[sort]."</a>|";
                echo "<a href=19-7.php?id=".$rows2[id].">评论（".$rows2[tbcount]."）</a>|";
                echo "<a href=19-7.php?id=".$rows2[id].">浏览（".$rows2[views]."）</a>";
                echo "<hr width=\"100\">";
            }
            echo "</td>";
            echo "</tr>";
        }
    }
    else                                         //如果没有任何参数
    {
        $sql="select id from $table_log where p_id=0"; //显示所有日志
        $result=mysql_query($sql,$link);
```

```
        $msg_count=mysql_num_rows($result);          //总条数
        $p_count=ceil($msg_count/10);                //总页数
        echo "<tr>";
        echo "<td bgcolor=\"#eeeeff\">全部日志</td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeeeff\">";
        if ($_GET["page"]==0 && !$_GET["page"])
        $page=1;
        else
        $page=$_GET[page];
        $s=($page-1)*10+1;
        $s=$s-1;
        $sql="select * from $table_log where p_id=0 order by id desc limit $s, 10";
        $result=mysql_query($sql,$link);
        $nums=mysql_num_rows($result);
        if($nums<1) echo "还没有任何日志记录！";    //没有日志时显示信息
        else                                         //如果有日志
        {
            while($rows=mysql_fetch_array($result))  //循环显示日志内容
            {
                echo "<a href=19-7.php?id=".$rows[id].">".$rows[title]."</a>";
                echo "<p>";
                echo $rows[date];
                echo "<p>";
                echo $rows[content];
                echo "<p>";
                echo "<a href=19-7.php?id=".$rows[id].">阅读全文</a>|";
                echo "<a href=19-6.php?sort=".$rows[sort].">类别：".$rows[sort]."</a>|";
                echo "<a href=19-7.php?id=".$rows[id].">评论（".$rows[tbcount]."）</a>|";
                echo "<a href=19-7.php?id=".$rows[id].">浏览（".$rows[views]."）</a>";
                echo "<hr width=\"100%\">";
            }
        }
        echo "</td>";
        echo "</tr>";
    }
}
else                                                 //如果有日期参数
{
    $y=$_GET[y];                                     //获取年
    $m=$_GET[m];                                     //获取月
    $d=$_GET[d];                                     //获取日
    $date=$y."年".$m."月".$d."日";                   //代入新的变量中
    echo "<tr>";
    echo "<td bgcolor=\"#eeeeff\">".$date."的日志</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td bgcolor=\"#eeeeff\">";                 //查找当天的日志
```

```
        $sql="select * from $table_log where date like '$date'and p_id=0";
        $result=mysql_query($sql,$link);
        $nums=mysql_num_rows($result);                    //获取当天日志数量
        if($nums<1) echo "当天没有任何日志记录！ ";        //如果不存在
        else                                              //如果存在
        {
            while($rows=mysql_fetch_array($result))       //循环显示
            {
                echo "<a href=19-7.php?id=".$rows[id].">".$rows[title]."</a>";
                echo "<p>";
                echo $rows[date];
                echo "<p>";
                echo $rows[content];
                echo "<p>";
                echo "<a href=19-7.php?id=".$rows[id].">阅读全文</a>|";
                echo "<a href=19-6.php?sort=".$rows[sort].">类别".$rows[sort]."</a>|";
                echo "<a href=19-7.php?id=".$rows[id].">评论（".$rows[tbcount]."）</a>|";
                echo "<a href=19-7.php?id=".$rows[id].">浏览（".$rows[views]."）</a>";
                echo "<hr width=\"100\">";
            }
        }
        echo "</td>";
        echo "</tr>";
    }
    if(!($_GET[y]and$_GET[m]and$_GET[d])&&!($_GET[sort]))
    {                                                     //显示分页连接
        echo "<tr>";
        echo "<td bgcolor=\"#eeeeff\">";
        $prev_page=$page-1;
        $next_page=$page+1;
        if ($page<=1){
            echo "第一页 | ";
        }
        else{
            echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
        }
        if ($prev_page<1){
        echo "上一页 | ";
        }
        else{
            echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
        }
        if ($next_page>$p_count){
            echo "下一页 | ";
        }
        else{
            echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
        }
        if ($page>=$p_count){
```

```
            echo "最后一页</p>\n";
        }
        else{
            echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
        }
        echo "</td>";
        echo "</tr>";
    }
    echo "</table>";
    echo "</td>";
    echo "</tr>";
    echo "</table>";
    ?>
```

将以上代码保存为“19-6.php”。查看日志的条件可以是某一类的所有日志，也可以是某一天所写的所有日志。以类别查看日志如图 19.5 所示（前提是管理员后台登录并添加相应类别及日志）。从图 19.5 可以发现，显示出了所有类别为 PHP 的日志。因为当前只有一篇类别为 PHP 日志，所以只显示一篇。

接下来选择左侧边的表单为 2006 年 10 月，再单击 18 日图标，相当于执行 19-6.php?y=2006&m=10&d=18 命令。即执行 19-6.php 同时代入年、月、日参数。执行结果如图 19.6 所示。

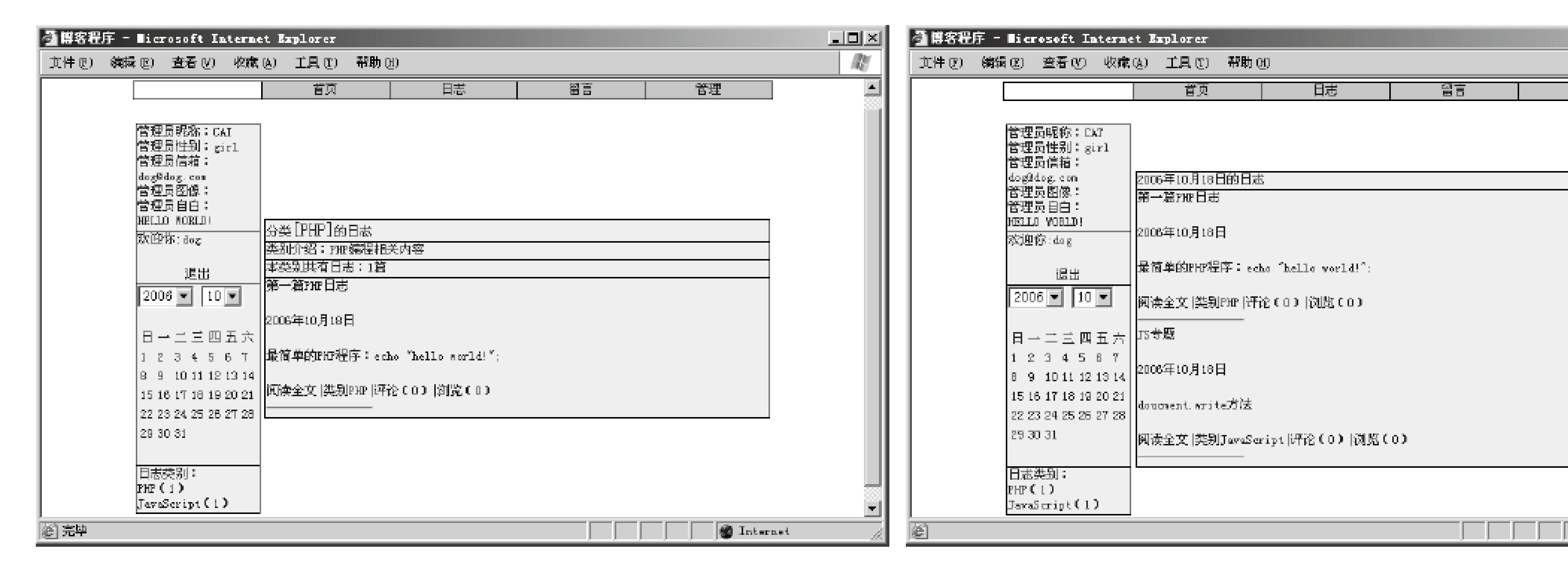


图 19.5　查看某一类别的日志执行结果　　　　图 19.6　查看某一天的日志执行结果

从以上两次执行结果可以看出，日志显示页面成功显示出了按照显示条件所列出的所有日志。

19.4.3　单条日志详细信息页面的创建

19.4.2 小节创建的日志显示页面，只显示相关日志，并不显示日志的详细情况及相关的评论。这一节来创建单条日志详细信息的页面。该页面不仅显示某一条日志的详细信息，而且还显示了所有对该日志的评论。代码如下：

```
<?
require "19-3.php";                                    //调用头文件
if(!$_POST[id])                                        //如果没有评论 ID 输入
```

```
{
echo "<table width=\"80%\">\n";
echo "<tr>";
echo "<td width=\"20%\">";
require "19-4.php";                                //调用侧边文件
echo "</td>";
echo "<td width=\"80%\">";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" align=\"center\" bgcolor=\"#000000\">\n";
require "19-1.php";
$link=mysql_connect($host,$user,$pass);            //连接主机
mysql_select_db($db_name,$link);                   //选择库
$sql="select * from $table_log where id='$_GET[id]' order by id desc";
$result=mysql_query($sql,$link);                   //发送显示某一条日志的 SQL 请求
$rows=mysql_fetch_array($result,$link);            //置入数组
echo "<tr bgcolor=\"#ffffff\">";
echo "<td>";
echo "<h3>".$rows[title]."</h3>";                  //显示标题
echo "</td>";
echo "</tr>";
echo "<tr bgcolor=\"#ffffff\">";
echo "<td>";
echo $rows[title];                                 //显示标题
echo "<p>";
echo $rows[date];                                  //显示日期
echo "<p>";
echo $rows[content];                               //显示内容
echo "<p>";
echo "<a href=19-6.php?sort=".$rows[sort].">类别： ".$rows[sort]."</a>|";
echo "评论（".$rows[tbcount]."）|";                 //显示评论数
echo "浏览（".$rows[views]."）";                    //显示浏览数
echo "<hr width=\"100%\">";
echo "以下为评论内容：<br>";                         //显示对该条日志的评论
$sql2="select * from $table_log where p_id='$_GET[id]'";
$result2=mysql_query($sql2,$link);                 //发送 SQL 请求
while($row2=mysql_fetch_array($result2,$link))     //循环显示评论
{
echo "标题：".$row2[title];                         //显示评论标题
echo "<p>";
echo "日期：".$row2[date];                          //显示评论日期
echo "<p>";
echo "内容：".$row2[content];                       //显示评论内容
echo "<hr width=\"80%\">";
}                                                  //下面一句每浏览一次浏览次数增加 1
$sql="update $table_log set views=views+1 where id='$_GET[id]'";
mysql_query($sql,$link);                           //发送自增 1 的 SQL 请求
echo "</td>";
echo "</tr>";
echo "</table>";
echo "<p>";
```

```
echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">";
echo "<form action=19-7.php method=post>";                  //以下为添加评论的表单
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "发表评论：";
echo "<input type=hidden name=id value=".$rows[id].">";
echo "<input type=hidden name=sort value=".$rows[sort].">";
echo "</td></tr>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "题目：<input type=text name=title>";
echo "</td></tr>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "评论内容：<br><textarea name=content rows=5 cols=40></textarea>";
echo "</td></tr>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "<input type=submit value=\"发表\">";
echo "</td></tr>";
echo "</form>";
echo "</table>";
echo "</td>";
echo "</tr>";
echo "</table>";
}
else                                                        //处理用户提交评论
{
$id=$_POST[id];                                             //获取各项参数
$title=$_POST[title];
$sort=$_POST[sort];
$content=$_POST[content];
$date=$date=date("Y 年 n 月 d 日");
if(!$_COOKIE["username"])                                   //如果没有用户登录
{
$username="匿名";                                           //匿名用户
}
else
{
$username=$_COOKIE["username"];                             //登录用户
}
require "19-1.php";                                         //调用配置文件
$link=mysql_connect($host,$user,$pass);
mysql_select_db($db_name,$link);
$sql="insert                                                                      into
$table_log(p_id,author,title,content,sort,date)values('$id','$username','$title','$content','$sort','$date')";
mysql_query($sql,$link);                                    //插入评论记录
$sql="update $table_log set tbcount=tbcount+1 where id='$id'";
mysql_query($sql,$link);                                    //评论数自增 1
echo "<meta http-equiv=\"refresh\" content=\"2; url=19-7.php?id=".$id." \">";
echo "</head>\n";
echo "<body>\n";
echo "添加成功评论，正在返回";
```

```
echo "</body>";
echo "</html>";
}
?>
```

保存以上代码为“19-7.php”。

单击某一条日志的标题，或者单击“阅读全文”链接，均可以进入查看该条日志详情的页面，相当于执行 19-7.php。如查看第一篇 PHP 日志（这都是在以管理员身份登录系统并添加相应日志后才能显示的），执行结果如图 19.7 所示。



图 19.7　查看日志详情执行结果

从图 19.7 可以发现，成功显示出了当前日志的详细内容及所有评论，并且可以通过该页面为日志添加评论。说明程序成功运行。

19.5　管 理 模 块

这一节来创建管理模块，管理模块不仅包括日志的管理，还包括用户信息的管理、留言管理、注册用户管理等内容。用户管理使每个用户可以修改自己的注册信息。而日志管理、留言管理、所用注册用户的管理则属于管理员特有的权限。下面就为读者作一一说明。

19.5.1　管理员/用户登录页面

这一小节来创建用户登录页，用户包括普通用户与管理员。管理员登录以后，可以对博客的整体情况进行全面的管理，如签写日志、分类管理、日志管理等内容。而普通用户登录之后则可以行使匿名用户所不具有的权限，例如对日志发表评论，浏览匿名用户不能浏览的日志等。不管何种用户都可以更改该用户注册的各项信息。

下面给出详细内容：

```
<?
if(!$_POST[username])
{
	require "19-3.php";	//调用头文件
	echo "<script language=\"javascript\">\n";
	echo "function juge(theForm)\n";
	echo "{\n";
	echo "\tif (theForm.username.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入用户名！\");\n";
	echo "\t\ttheForm.admin.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.password.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入密码！\");\n";
	echo "\t\ttheForm.pass.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "}\n";
	echo "</script>\n";
	echo "<center>\n";
	echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\" width=\"80%\">\n";
	echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
	echo "<tr bgcolor=\"#cccc99\">\n";
	echo "<td colspan=\"2\" align=\"center\">管理员/用户登录</td>\n";
	echo "</tr>\n";
	echo "<tr   bgcolor=\"#cccc99\">\n";
	echo "<td>输入用户名</td>\n";
	echo "<td><input type=\"text\" name=\"username\"></td>\n";
	echo "</tr>\n";
	echo "<tr bgcolor=\"#cccc99\">\n";
	echo "<td>输入密码：</td>\n";
	echo "<td><input type=\"password\" name=\"password\" size=\"21\"></td>\n";
	echo "</tr>\n";
	echo "<tr bgcolor=\"#cccc99\">\n";
	echo "<td colspan=\"2\"><center><input type=\"submit\" value=\"登录\"></center></td>\n";
	echo "</tr>\n";
	echo "</form>\n";
	echo "</table>\n";
	echo "</center>\n";
	echo "</body>\n";
	echo "<html>\n";
}
else
{
	$username=$_POST[username];	//获取用户名
	$password=md5($_POST[password]);	//获取密码，并用 MD5 加密
	require "19-1.php";	//调用配置文件
```

```
$link=mysql_connect($host,$user,$pass);            //连接主机
mysql_select_db($db_name,$link);                   //选择数据库
$sql="select * from $table_user where username='$username' and password='$password'";
$result=mysql_query($sql,$link) or die(mysql_error());
$row=mysql_num_rows($result);                      //获取符合的用户数量
if($row<1)                                         //如果符合的数量为 0
{
    require "19-3.php";                            //调用头文件
    echo "<meta http-equiv=\"refresh\" content=\"2; url=19-8.php\">";
    echo "用户名或者密码错误两秒后返回！";          //显示错误信息
}
else                                               //如果存在符合条件的用户
{
    setcookie("username","$username");             //注册 COOKIE
    require "19-3.php";                            //调用头文件
    echo "<table width=\"80%\">\n";
    echo "<tr>";
    echo "<td width=\"20%\">";
    $rows=mysql_fetch_array($result);              //用户信息置入数组
    echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">\n";
    echo "<tr>";
    echo "<td bgcolor=\"#ffeeee\">";
    echo "<a href=\"19-9.php\" target=\"fram\">用户信息</a>";
    echo "</td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td bgcolor=\"#eeffee\">";
    echo "<a href=\"19-10.php\" target=\"fram\">更改密码</a>";
    echo "</td>";
    echo "</tr>";
    if($rows[admin]=="1")                          //如果用户为管理员，则显示更多操作
    {
        echo "<tr>";
        echo "<td bgcolor=\"#eeffee\"><a href=19-11.php target=\"fram\">分类管理</a></td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeffee\"><a href=19-16.php target=\"fram\">添加分类</a></td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeffee\"><a href=19-12.php target=\"fram\">发表日志</a></td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeffee\"><a href=19-13.php target=\"fram\">日志管理</a></td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeffee\"><a href=19-14.php target=\"fram\">留言管理</a></td>";
        echo "</tr>";
        echo "<tr>";
        echo "<td bgcolor=\"#eeffee\"><a href=19-15.php target=\"fram\">用户管理</a></td>";
```

```
                echo "</tr>";
            }
            echo "</table>";
            echo "</td>";
            echo "<td width=\"80%\">";
            echo "<iframe name=\"fram\" src=\"19-9.php\" width=\"476\" height=\"350\" frameBorder=\"0\">";
            echo "</td>";
            echo "</tr>";
            echo "</table>";
        }
}
?>
```

把以上代码保存为“19-8.php”。现在就可以用管理员的身份登录该页，以便执行管理分类、添加分类、签写日志、管理日志、管理留言、管理用户等操作了。

首次执行 19-8.php，其执行结果如图 19.8 所示。按照要求输入用户名及密码就可以成功登录系统。成功登录后的界面如图 19.9 所示。



图 19.8　管理/用户登录页执行结果

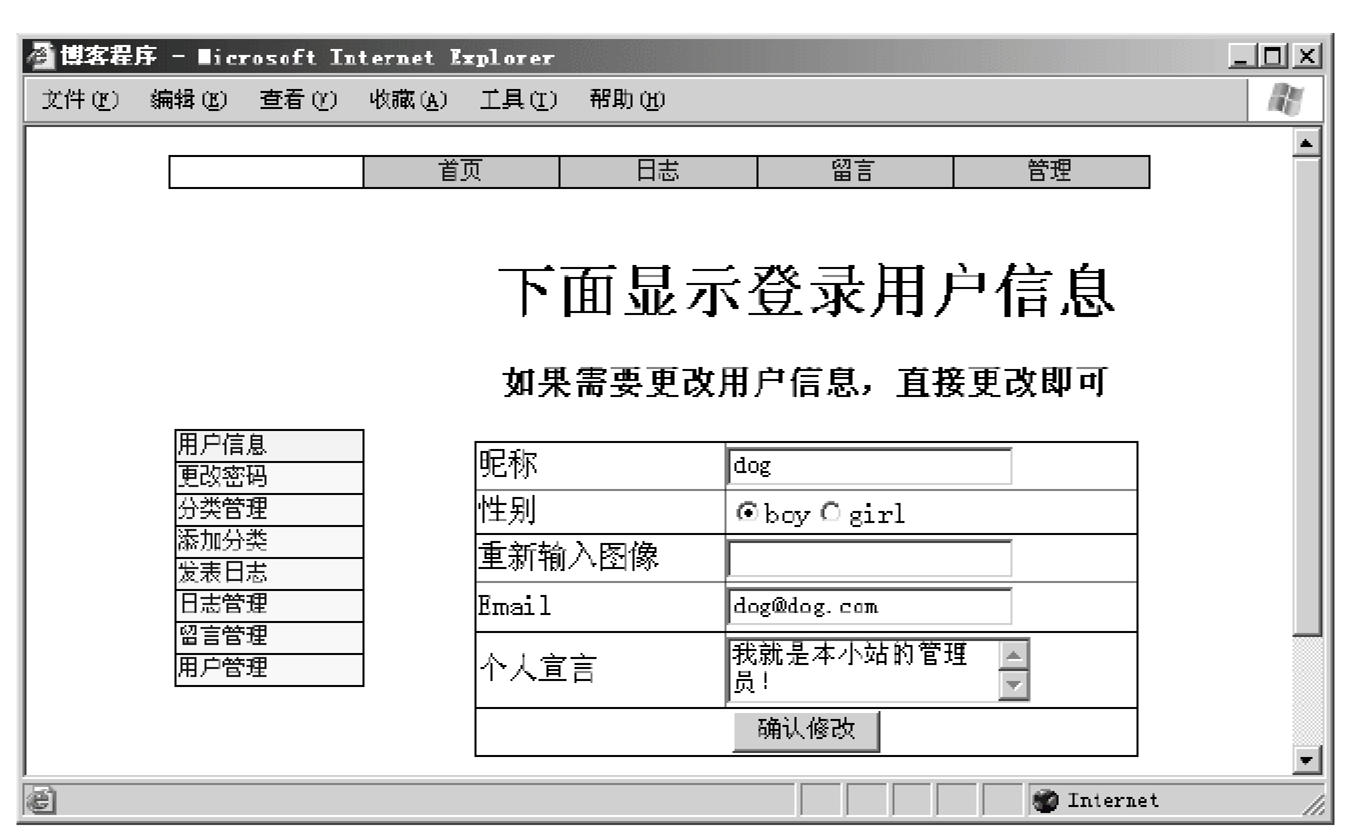


图 19.9　以管理员身份登录后执行结果

19.5.2　更改用户注册信息

无论是管理员或者是普通注册用户都有权更改已经注册的本用户信息，这一节就来创建更改注册用户信息的页面。

具体内容参看以下代码：

```
<?php
if(!$_COOKIE[username])                    //如果没有登录用户
{
    "没有用户登录！";
}
else
{
    if(!$_POST[nickname])                  //如果没有发送表单变量，则显示 HTML
```

```
    {
        echo "<html>\n";
        echo "<head>\n";
        echo "<title>博客程序——显示登录用户信息</title>\n";
        echo "</head>\n";
        echo "<body>\n";
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.nickname.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"匿称必须要填！\");\n";
        echo "\t\ttheForm.nickname.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "</script>\n";
        require "19-1.php";
        $link=mysql_connect($host,$user,$pass);
        mysql_select_db($db_name,$link);
        $sql="select * from $table_user where username='$_COOKIE[username]'";
        $result=mysql_query($sql,$link);
        $rows=mysql_fetch_array($result);         //登录用户信息置入数组
        echo "<center>\n";
        echo "<h1>下面显示登录用户信息</h1>";
        echo "<h3>如果需要更改用户信息，直接更改即可</h3>";
        echo    "<table    width=\"80%\"    cellpadding=\"1\"    cellspacing=\"1\"    align=\"center\"
bgcolor=\"#000000\">\n";
        echo "<form action=19-9.php method=post onsubmit=\"return juge(this)\">";
        echo "<tr>\n";
        echo "<td bgcolor=\"#ffffff\">昵称</td>\n";
        echo "<td bgcolor=\"#ffffff\"><input type=\"text\" name=\"nickname\" value=\"";
        echo $rows[nickname];
        echo "\" ></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td bgcolor=\"#ffffff\">性别</td>\n";
        echo "<td bgcolor=\"#ffffff\">";
        echo "<input type=radio name=sex value=boy ";
        if($rows[sex]=="boy") echo "checked";
        echo " >boy";
        echo "<input type=radio name=sex value=girl ";
        if($rows[sex]=="girl") echo "checked";
        echo " >girl";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td bgcolor=\"#ffffff\">重新输入图像</td>\n";
        echo "<td bgcolor=\"#ffffff\"><input type=\"text\" name=\"photo\" value=\"";
```

```
            echo "\" ></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td bgcolor=\"#ffffff\">Email</td>\n";
            echo "<td bgcolor=\"#ffffff\"><input type=\"text\" name=\"email\" value=\"";
            echo $rows[email];
            echo "\" ></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td bgcolor=\"#ffffff\">个人宣言</td>\n";
            echo "<td bgcolor=\"#ffffff\">";
            echo "<textarea name=\"description\">".$rows[description]."</textarea>";
            echo "</td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td bgcolor=\"#ffffff\" colspan=\"2\"><center><input type=submit value=\" 确 认 修 改
\"></center></td>\n";
            echo "</tr>\n";
            echo "</form>";
            echo "</table>\n";
        }
        else                                                    //如果发送表单
        {
            $nickname=$_POST[nickname];                         //获取表单变量
            $sex=$_POST[sex];
            $photo=$_POST[photo];
            $email=$_POST[email];
            $description=$_POST[description];
            require("19-1.php");
            $link=mysql_connect($host,$user,$pass);
            mysql_select_db($db_name,$link);
            $sql="update                    $table_user                    set
nickname='$nickname',sex='$sex',photo='$photo',email='$email',description='$description'    where
username='$_COOKIE[username]'";
            $result=mysql_query($sql,$link);                    //发送修改表内容的 SQL 请求
            echo "<html>\n";
            echo "<head>\n";
            echo "<title>博客程序——用户登录</title>\n";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=19-9.php\">";
            echo "</head>\n";
            echo "<body>\n";
            echo "用户信息更改成功！两秒后返回";
            echo "</body>";
            echo "</html>";
        }
}
?>
```

把以上代码保存为“19-9.php”。因为该页面是以框架形式包含在 19-8.php 中的，所以从图 19.9

中可以了解到该 PHP 文件的执行结果，即图 19.9 的右下方所显示的内容。实际上这里就相当于执行了 19-9.php。这里可以更改用户的昵称、性别、电子信箱、个人宣言等信息。需要注意的是管理员的个人信息将显示在首页页面上。在这里，把管理员的昵称由 dog 改为 CAT，个人宣言改为“HELLO WORLD！”以查看 19-9.php 的执行结果。进行相应修改并单击“确认修改”按钮之后，执行结果将如图 19.10 所示。



图 19.10　更改过用户信息之后的执行结果

在执行过这一步的操作之后，单击最上面的首页链接，回到首页将会看到显示管理员信息的区域已经显示了新的信息。

19.5.3　更改注册用户密码页面

因为用户密码关系着用户账号的安全，所以不能跟用户其他信息一样，而是要单独修改。这样可以保护密码的安全，并且密码都是把用户的输入经过了 md5 的加密。就算得到了明码，也不会知道密码的内容。

具体内容参考以下代码：

```
<?php
if(!$_COOKIE[username])                         //如果没有登录用户，则显示信息
{
    "没有用户登录！";
}
else                                            //如果用户已经登录
{
    if(!$_POST[old_pass])                       //如果没有表单发送，则显示 HTML
    {
        echo "<html>\n";
        echo "<head>\n";
        echo "<title>博客程序——更改登录用户密码</title>\n";
        echo "</head>\n";
        echo "<body>\n";
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.old_pass.value == \"\")\n";
```

```
echo "\t{\n";
echo "\t\talert(\"请输入原始密码！\");\n";
echo "\t\ttheForm.old_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.new_pass.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入新密码！\");\n";
echo "\t\ttheForm.new_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.new_pass.value.length < 8 )\n";
echo "\t{\n";
echo "\t\talert(\"密码至少要 8 位！\");\n";
echo "\t\ttheForm.new_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.re_pass.value !=theForm.new_pass.value)\n";
echo "\t{\n";
echo "\t\talert(\"确认密码与密码不一致！\");\n";
echo "\t\ttheForm.re_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<h3>输入原始密码及新密码</h3>";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\"
bgcolor=\"#000000\">\n";
echo "<form action=19-10.php method=post onsubmit=\"return juge(this)\">";
echo "<tr>\n";
echo "<td bgcolor=\"#ffffff\">原始密码：</td>\n";
echo "<td bgcolor=\"#ffffff\"><input type=\"password\" name=\"old_pass\" value=\"";
echo "\" ></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td bgcolor=\"#ffffff\">新密码：</td>\n";
echo "<td bgcolor=\"#ffffff\"><input type=\"password\" name=\"new_pass\" value=\"";
echo "\" ></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td bgcolor=\"#ffffff\">确认新密码：</td>\n";
echo "<td bgcolor=\"#ffffff\"><input type=\"password\" name=\"re_pass\" value=\"";
echo "\" ></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td bgcolor=\"#ffffff\" colspan=\"2\"><center><input type=submit value=\" 确 认 修 改
\"></center></td>\n";
echo "</tr>\n";
```

```
		echo "</form>";
		echo "</table>\n";
	}
	else									//如果发送表单变量
	{
		$old_pass=md5($_POST[old_pass]);			//获取表单变量
		$new_pass=$_POST[new_pass];
		require "19-1.php";
		$link=mysql_connect($host,$user,$pass);
		mysql_select_db($db_name,$link);
		$sql="select * from $table_user where username='$_COOKIE[username]' and 
password='$old_pass'";
		$reslut=mysql_query($sql,$link);			//发送验证用户身份的 SQL 请求
		$row=mysql_num_rows($result);				//符合条件的记录数
		if($row<1)								//如果不存在符合条件的记录，则显示信息
		{
			echo "<html>\n";
			echo "<head>\n";
			echo "<title>博客程序——用户登录</title>\n";
			echo "<meta http-equiv=\"refresh\" content=\"2; url=19-10.php\">";
			echo "</head>\n";
			echo "<body>\n";
			echo "输入的密码错误！ 或者用户不存在！ 两秒后返回!";
			echo "</body>";
			echo "</html>";
		}
		else									//如果旧密码正确
		{
			$sql="update $table_user set password='$new_pass' where username='$_COOKIE[username]' 
and password='$old_pass'";
			$result=mysql_query($sql,$link);		//发送更改旧密码为新密码的 SQL 请求
			echo "<html>\n";
			echo "<head>\n";
			echo "<title>博客程序——用户登录</title>\n";
			echo "<meta http-equiv=\"refresh\" content=\"2; url=19-10.php\">";
			echo "</head>\n";
			echo "<body>\n";
			echo "更改成功！ 以后请使用新密码登录!";
			echo "</body>";
			echo "</html>";
		}
	}
}
?>
```

把以上代码保存为“19-10.php”，以便在用户需要对密码进行更改时调用该文件。同 19-9.php 一样，该页面也是以框架形式包含在 19-8.php 中的。由于修改密码操作与修改用户信息操作基本类似，这里就不再给执行结果图了。

用户信息更改及用户密码更改是所有用户都拥有的权限。下面几节的操作就是只有管理员才有的

权限，如日志管理、分类管理、留言管理和用户管理等。从 19.5.4 小节开始一个一个地创建这些页面。

19.5.4　修改已经存在的日志类别

日志的类别是日志的一个重要属性，所以对类别的管理也很重要。这一小节来创建类别管理的页面。和签写新日志功能一样，这当然也是管理员才有的超级权限，所以在用户调用该页时要对用户的身份进行判断。

具体内容参看以下代码：

```
<?
if($_COOKIE[username])                              //如果用户已经登录
{
    require "19-1.php";                             //调用配置文件
    $link=mysql_connect($host,$user,$pass);
    mysql_select_db($db_name,$link);
    $sql="select * from $table_user where username='$_COOKIE[username]'";
    $result=mysql_query($sql,$link);                //发送获取用户信息的 SQL 请求
    $row=mysql_fetch_array($result);                //用户信息置入数组
    if ($row[admin]=="1")                           //如果用户为管理员，则执行以下操作
    {
        if(!$_POST[action])                         //如果没有发送表单变量，则显示 HTML
        {
            echo "<center>\n";
            echo "<h3>修改类别</h3>";
            echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\"
width=\"100%\">\n";
            echo "<form method=\"post\" action=\"$PATH_INFO\">\n";
            $sql="select * from $table_sort";       //显示所有已经存在的类别
            $result=mysql_query($sql,$link);        //发送 SQL 请求
            echo "<tr>\n";
            echo "<td bgcolor=\"#ccffcc\"> 类 名 </td><td bgcolor=\"#ccffcc\"> 类 说 明 </td><td
bgcolor=\"#ccffcc\">操作类型</td>\n";
            echo "</tr>\n";
            while($rows=mysql_fetch_array($result)) //循环显示所有类
            {
                echo "<tr>";
                echo "<input type=hidden name=id[$i] value=".$rows[id].">";
                echo "<input type=hidden name=o_sort[$i] value=".$rows[sortname].">";
                echo "<td bgcolor=\"#eeeeff\"><input type=text value=".$rows[sortname]."
name=sortname[$i]></td><td bgcolor=\"#eeeeff\"><textarea name=description[$i] rows=3
cols=20>".$rows[description]."</textarea></td>";
                echo "<td bgcolor=\"#eeeeff\"><input type=radio name=action[$i] value=del checked>删除
<br><input type=radio name=action[$i] value=edit>修改<br></td>";
                echo "</tr>";
                $i++;
            }
            echo "<tr>";
```

```
                echo "<td colspan=\"3\" bgcolor=\"#eeffee\"><center>";
                echo "<input type=submit value=\"确认提交\">";
                echo "</center></td>";
                echo "</form>\n";
                echo "</table>\n";
                echo "</center>\n";
                echo "</body>\n";
                echo "<html>\n";
            }
            else                                        //如果发送表单变量，则执行操作
            {
                require "19-1.php";                     //调用配置文件
                $link=mysql_connect($host,$user,$pass); //连接 MYSQL 主机
                mysql_select_db($db_name,$link);        //选择数据库
                for($i=0;$i<count($_POST[id]);$i++)     //根据操作类型循环执行操作
                {
                    $temp1=$_POST[id][$i];              //循环获取表单变量
                    $temp2=$_POST[sortname][$i];
                    $temp3=$_POST[description][$i];
                    $temp4=$_POST[o_sort][$i];
                    if($_POST[action][$i]=="del")       //如果操作类型为删除
                    {
                        $sql="delete from $table_sort where id='$temp1'";
                        $sql2="delete from $table_log where sort='$temp2'";
                    }
                    else                                //如果操作类型为修改
                    {
                        $sql="update   $table_sort   set   sortname='$temp2',description='$temp3'   where
id='$temp1'";
                        $sql2="update $table_log set sort='$temp2' where sort='$temp4'";
                    }
                    mysql_query($sql);                  //发送对类的相关操作的 SQL 请求
                    mysql_query($sql2);                 //发送对日志相关操作的 SQL 请求
                }
                echo "<html>\n";
                echo "<head>\n";
                echo "<title>博客程序</title>\n";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=19-11.php\">";
                echo "</head>\n";
                echo "<body>\n";
                echo "处理成功，正在返回";
                echo "</body>";
                echo "</html>";
            }
        }
        else                                            //如果用户不是管理员
        {
            echo "普通用户没有该权限!";                   //显示信息
        }
```

```
}
else                                                  //如果用户没有登录
{
    echo "你还没有登录,点<a href=19-8.php>这里</a>进行登录!";
}
?>
```

保存以上代码为“19-11.php”。以方便管理员通过该页面来管理已经存在的日志类别。用户以管理员身份登录系统，并执行该操作时，其执行结果如图 19.11 所示。



图 19.11　分类管理执行结果

从图 19.11 可以发现，现在共有两个类别，这里对默认类别进行修改操作，先选择操作类型为修改，类名文本框中输入 PHP，类说明文本框中输入“PHP 编程相关内容”。对第 2 个类不进行操作，则要选择操作类型为修改，其中各项内容保持不变即可。然后，单击“确认提交”按钮，执行结果如 19.12 所示。



图 19.12　类别修改过之后的执行结果

通过图 19.12 可以发现，默认添加的类别，已经被改成了 PHP，说明修改操作成功执行。

19.5.5　签写新的日志页面

由于刚建成的系统中并没有任何日志。而博客程序必须要使管理员具有写日志这项权限，这也是最基本的功能。这一小节来创建管理员签写新日志的页面。

具体内容参看如下代码：

```
<?
if($_COOKIE[username])                                  //如果用户已经登录
{
        require "19-1.php";                             //调用配置文件
        $link=mysql_connect($host,$user,$pass);         //连接主机
        mysql_select_db($db_name,$link);                //选择数据库
        $sql="select * from $table_user where username='$_COOKIE[username]'";
        $result=mysql_query($sql,$link);                //发送获取用户信息的 SQL 请求
        $row=mysql_fetch_array($result);                //用户信息置入数组
        if ($row[admin]=="1")                           //如果用户为管理员，则执行以下操作
        {
                if(!$_POST[content])                    //如果没有发送表单变量，则显示 HTML
                {
                        echo "<script language=\"javascript\">\n";
                        echo "function juge(theForm)\n";
                        echo "{\n";
                        echo "\tif (theForm.title.value == \"\")\n";
                        echo "\t{\n";
                        echo "\t\talert(\"请输入日志标题！ \");\n";
                        echo "\t\ttheForm.title.focus();\n";
                        echo "\t\treturn (false);\n";
                        echo "\t}\n";
                        echo "\tif (theForm.content.value == \"\")\n";
                        echo "\t{\n";
                        echo "\t\talert(\"请输入日志内容！ \");\n";
                        echo "\t\ttheForm.content.focus();\n";
                        echo "\t\treturn (false);\n";
                        echo "\t}\n";
                        echo "}\n";
                        echo "</script>\n";
                        echo "<center>\n";
                        echo    "<table    cellpadding=\"1\"    cellspacing=\"1\"    align=\"center\"    bgcolor=\"#000000\"
width=\"100%\">\n";
                        echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
                        echo "<tr bgcolor=\"#cccc99\">\n";
                        echo "<td colspan=\"2\" align=\"center\">签写新的日志</td>\n";
                        echo "</tr>\n";
                        echo "<tr    bgcolor=\"#cccc99\">\n";
                        echo "<td>日志标题</td>\n";
                        echo "<td><input type=\"text\" name=\"title\"></td>\n";
                        echo "</tr>\n";
                        echo "<tr    bgcolor=\"#cccc99\">\n";
                        echo "<td>所属类别</td>\n";
                        echo "<td><select name=\"sort\" size=\"1\">";
                        $sql="select sortname from $table_sort";
                        $result=mysql_query($sql,$link);
                        while($rows=mysql_fetch_array($result))
                        {
                                echo "<option value=\"";
```

```
                echo $rows[0];
                echo "\">";
                echo $rows[0];
            }                                           //通过循环显示所有存在类别
            echo "</select></td>\n";
            echo "</tr>\n";
            echo "<tr bgcolor=\"#cccc99\">\n";
            echo "<td>日志内容：</td>\n";
            echo "<td><textarea rows=\"10\" cols=\"40\" name=\"content\"></textarea></td>\n";
            echo "</tr>\n";
            echo "<tr bgcolor=\"#cccc99\">\n";
            echo "<td>是否可见：</td>\n";
            echo  "<td><input  type=radio  name=\"hide\"  value=n  checked> 是 <input  type=radio
name=\"hide\" value=y>否</td>\n";
            echo "</tr>\n";
            echo "<tr bgcolor=\"#cccc99\">\n";
            echo "<td colspan=\"2\"><center><input type=\"submit\" value=\"确认提交\"></center></td>\n";
            echo "</tr>\n";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "<html>\n";
        }
        else                                            //如果发送表单变量，则执行操作
        {
            $title=$_POST[title];                       //获取表单变量
            $content=$_POST[content];
            $hide=$_POST[hide];
            $sort=$_POST[sort];
            $date=date("Y 年 n 月 d 日");                //获取当前日期
            require "19-1.php";
            $link=mysql_connect($host,$user,$pass);
            mysql_select_db($db_name,$link);
            $sql="insert                                                                    into
$table_log(title,content,sort,author,hide,date)values('$title','$content','$sort','$_COOKIE[username]','$hide','$dat
e')";
            mysql_query($sql,$link);                    //发送插入新日志记录的 SQL 请求
            $sql2="update $table_sort set sortnum=sortnum+1 where sortname='$sort'";
            mysql_query($sql2,$link);                   //发送更改类别数量的 SQL 请求
            echo "<html>\n";
            echo "<head>\n";
            echo "<title>博客程序</title>\n";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=19-12.php\">";
            echo "</head>\n";
            echo "<body>\n";
            echo "添加成功，正在返回";
            echo "</body>";
            echo "</html>";
```

```
            }
        }
        else                                                //如果用户不是管理员
        {
            echo "普通用户没有该权限!";
        }
    }
    else                                                    //如果没有登录用户
    {
        echo "你还没有登录,点<a href=19-8.php>这里</a>进行登录!";
    }
    ?>
```

把上面的代码保存为“19-12.php”。这样，当以管理员身份登录系统以后，就可以调用该页面来写新的日志，或者叫“写博客”了。管理员登录之后，应该能够签写新的日志。现在单击图 19.12 左侧边的“发表日志”链接，相当于执行 19-12.php。执行结果如图 19.13 所示。

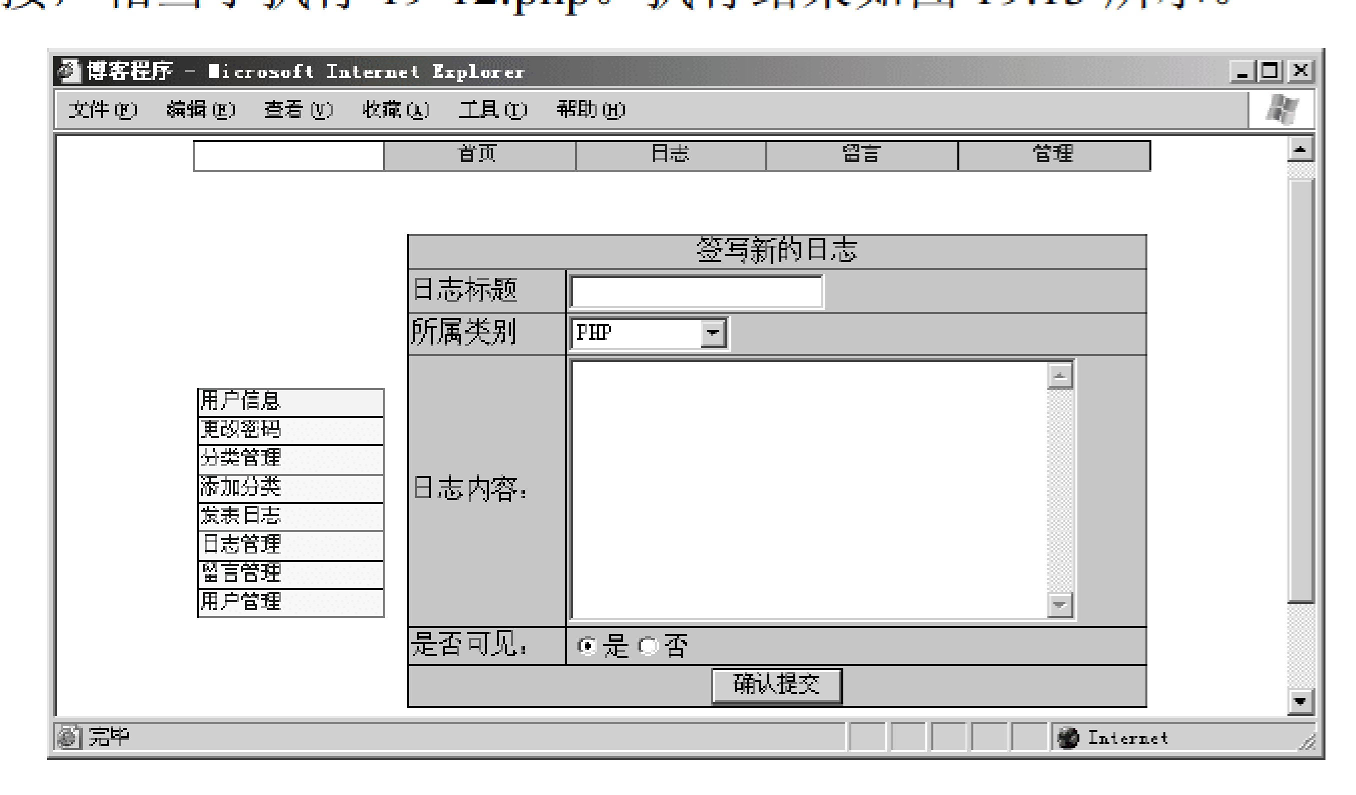


图 19.13　签写新的日志执行结果

为了方便以后测试，这里可以输入几条以作为测试之用。

在日志标题文本框中输入“第一篇 PHP 日志”，类别选择“PHP”选项，日志内容，随便填写一些内容以作为测试，然后可以再输入一条类别为 JavaScript 的记录。

发表日志之后，当时发现不了结果，因为操作成功之后都又跳转到了签写新日志页。不过通过下面的操作就能看到所写的日志了。

19.5.6　已经存在的日志管理页

写过的日志，有时需要进行修改，另外觉得没有用的日志也要进行删除。所有这些操作都要有管理权限的管理员才能进行。这一小节就来创建日志的管理页。具体内容请参考以下代码：

```
    <?
    if($_COOKIE[username])                                  //如果没有登录用户
    {
        require "19-1.php";                                 //调用配置文件
        $link=mysql_connect($host,$user,$pass);             //连接主机
```

```
    mysql_select_db($db_name,$link);                    //选择数据库
    $sql="select * from $table_user where username='$_COOKIE[username]'";
    $result=mysql_query($sql,$link);                    //发送验证用户身份的 SQL 请求
    $row=mysql_fetch_array($result);                    //用户信息置入数组
    if ($row[admin]=="1")                               //如果用户是管理员，则执行以下操作
    {
        if(!$_POST[action])                             //如果没有发送表单变量，则显示 HTML
        {
            echo "<center>\n";
            echo "<h3>日志管理</h3>";
            echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\" 
width=\"100%\">\n";
            echo "<form method=\"post\" action=\"$PATH_INFO\">\n";
            $sql="select * from $table_log";
            $result=mysql_query($sql,$link);
            echo "<tr>\n";
            echo "<td bgcolor=\"#ccffcc\">id</td><td bgcolor=\"#ccffcc\">标题</td><td bgcolor=\"#ccffcc\">
内容</td><td bgcolor=\"#ccffcc\">操作类型</td>\n";
            echo "</tr>\n";
            while($rows=mysql_fetch_array($result))
            {
                echo "<tr>";
                echo "<input type=hidden name=id[$i] value=".$rows[id].">";
                echo "<td bgcolor=\"#eeeeff\">".$rows[id]."</td>";
                echo "<td bgcolor=\"#eeeeff\"><input type=text value=\"".$rows[title]."\" name=title[$i] 
size=10></td>";
                echo "<td bgcolor=\"#eeeeff\"><textarea name=content[$i] rows=3 
cols=20>".$rows[content]."</textarea></td>";
                echo "<td bgcolor=\"#eeeeff\"><input type=radio name=action[$i] value=del checked=1>删
除<br><input type=radio name=action[$i] value=edit>修改<br></td>";
                echo "</tr>";
                $i++;
            }                                           //循环显示日志内容及操作
            echo "<tr>";
            echo "<td colspan=\"4\" bgcolor=\"#eeffee\"><center>";
            echo "<input type=submit value=\"确认提交\">";
            echo "</center></td>";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "<html>\n";
        }
        else                                            //如果表单变量已经发送
        {
            require "19-1.php";                         //调用配置文件
            $link=mysql_connect($host,$user,$pass); //连接主机
            mysql_select_db($db_name,$link);            //选择数据库
            for($i=0;$i<count($_POST[action]);$i++)
            {
```

```
                    $temp1=$_POST[id][$i];              //循环获取表单变量
                    $temp2=$_POST[title][$i];
                    $temp3=$_POST[content][$i];
                    if($_POST[action][$i]=="del")        //如果操作类型为删除
                    {
                        $sql="delete from $table_log where id='$temp1'";
                    }
                    else                                 //如果操作类型为修改
                    {
                        $sql="update $table_log set title='$temp2',content='$temp3' where id='$temp1'";
                    }
                    mysql_query($sql);                   //执行删除或者修改操作
                }
                echo "<html>\n";
                echo "<head>\n";
                echo "<title>博客程序</title>\n";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=19-13.php\">";
                echo "</head>\n";
                echo "<body>\n";
                echo "处理成功，正在返回";
                echo "</body>";
                echo "</html>";
            }
        }
        else                                             //如果用户不是管理员，则显示信息
        {
            echo "普通用户没有该权限!";
        }
    }
    else                                                 //如果用户没有登录，则显示信息
    {
        echo "您还没有登录,点<a href=19-8.php>这里</a>进行登录!";
    }
    ?>
```

把以上代码保存为“19-13.php”。单击图 19.13 左侧的“日志管理”链接，相当于执行 19-13.php，其执行结果如图 19.14 所示。

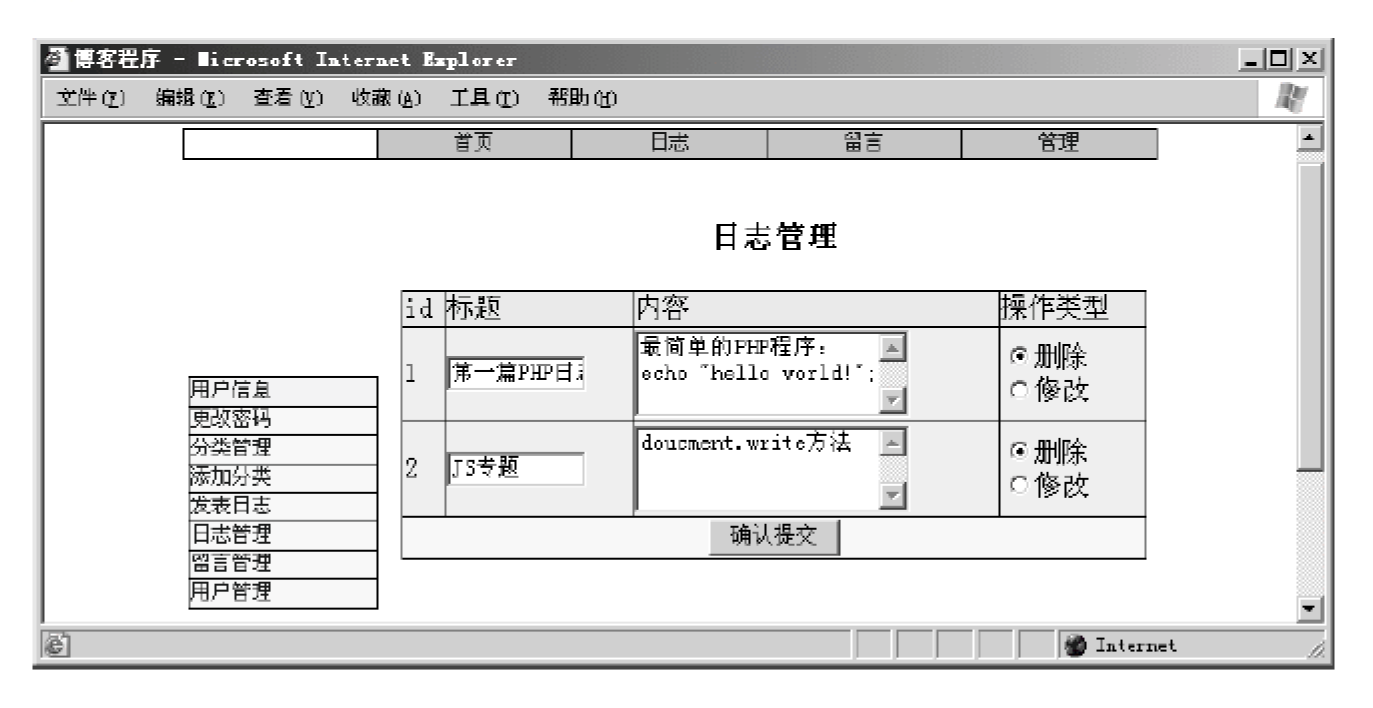


图 19.14　日志管理页面执行结果

这里显示出了已经存在的日志，可以修改日志的标题和内容，也可以选择删除某一条日志，按需要进行相关操作即可。如果只对某一项进行操作，其他项保持不变，则操作类型要选择修改，其他的如标题与内容保持原样即可。

19.5.7　留言的管理

相对于日志，留言是一个更为开放的窗口，里面可能充斥着大量的有用的或者没有用的信息，甚至包括乱七八糟的广告或者带毒的网页的链接。对这些内容要进行修改或者删除。这一小节来创建留言管理的页面。和其他的管理功能一样，只有管理员才有权限，才能在用户调用该页时要对用户的身份进行判断。

具体内容参看以下代码：

```
<?
if($_COOKIE[username])                                  //如果用户已经登录
{
    require "19-1.php";                                 //调用配置文件
    $link=mysql_connect($host,$user,$pass);
    mysql_select_db($db_name,$link);
    $sql="select * from $table_user where username='$_COOKIE[username]'";
    $result=mysql_query($sql,$link);
    $row=mysql_fetch_array($result);                    //用户信息置入数组
    if ($row[admin]=="1")                               //用户为管理员
    {
        if(!$_POST[action])                             //未发送表单变量时显示 HTML
        {
            echo "<center>\n";
            echo "<h3>留言管理</h3>";
            echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\" width=\"100%\">\n";
            echo "<form method=\"post\" action=\"$PATH_INFO\">\n";
            $sql="select * from $table_gbook";
            $result=mysql_query($sql,$link);
            echo "<tr>\n";
            echo "<td bgcolor=\"#ccffcc\">作 者</td><td bgcolor=\"#ccffcc\">标 题</td><td bgcolor=\"#ccffcc\">内容</td><td bgcolor=\"#ccffcc\">操作类型</td>\n";
            echo "</tr>\n";
            while($rows=mysql_fetch_array($result))
            {
                echo "<tr>";
                echo "<input type=hidden name=id[$i] value=".$rows[id].">";
                echo "<td bgcolor=\"#eeeeff\">".$rows[author]."</td><td bgcolor=\"#eeeeff\"><input type=text value=".$rows[title]." name=title[$i] size=6></td><td bgcolor=\"#eeeeff\"><textarea name=content[$i] rows=3 cols=20>".$rows[content]."</textarea></td>";
                echo "<td bgcolor=\"#eeeeff\"><input type=radio name=action[$i] value=del checked>删除<br><input type=radio name=action[$i] value=edit>修改<br></td>";
                echo "</tr>";
```

```
                $i++;
            }                                               //循环显示留言内容
            echo "<tr>";
            echo "<td colspan=\"4\" bgcolor=\"#eeffee\"><center>";
            echo "<input type=submit value=\"确认提交\">";
            echo "</center></td>";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "<html>\n";
        }
        else                                                //如果已经发送表单变量
        {
            require "19-1.php";
            $link=mysql_connect($host,$user,$pass);
            mysql_select_db($db_name,$link);
            for($i=0;$i<count($_POST[id]);$i++)
            {
                $temp1=$_POST[id][$i];                      //循环获取表单变量
                $temp2=$_POST[title][$i];
                $temp3=$_POST[content][$i];
                if($_POST[action][$i]=="del")               //如果操作类型为删除
                {
                    $sql="delete from $table_gbook where id='$temp1'";
                }
                else                                        //如果操作类型为修改
                {
                    $sql="update $table_gbook set title='$temp2',content='$temp3' where id='$temp1'";
                }
                mysql_query($sql);                          //执行操作 SQL 语句
            }
            echo "<html>\n";
            echo "<head>\n";
            echo "<title>博客程序</title>\n";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=19-14.php\">";
            echo "</head>\n";
            echo "<body>\n";
            echo "处理成功，正在返回";
            echo "</body>";
            echo "</html>";
        }
    }
    else                                                    //如果用户不是管理员
    {
        echo "普通用户没有该权限!";
    }
}
else                                                        //如果用户没有登录
```

```
{
        echo "你还没有登录,点<a href=19-8.php>这里</a>进行登录!";
}
?>
```

把以上代码保存为“19-14.php”，由于该功能与日志管理类似，所以也不再给出执行效果图了。

19.5.8　注册用户的管理

对注册用户进行管理也是管理工作中一个很重要的环节。通过该页面，管理员可以修改注册用户的信息（但是一般情况下，不主张这么做），甚至可以从数据库中删除某一个注册用户。

具体内容参看如下代码：

```
<?
if($_COOKIE[username])                                      //如果用户已经登录
{
        require "19-1.php";                                 //调用配置文件
        $link=mysql_connect($host,$user,$pass);             //连接主机
        mysql_select_db($db_name,$link);                    //选择数据库
        $sql="select * from $table_user where username='$_COOKIE[username]'";
        $result=mysql_query($sql,$link);                    //验证用户身份
        $row=mysql_fetch_array($result);
        if ($row[admin]=="1")                               //如果用户为管理员
        {
                if(!$_POST[action])                         //如果没有发送表单变量，则显示 HTML
                {
                        echo "<center>\n";
                        echo "<h3>用户管理</h3>";
                        echo "<table cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\"
width=\"100%\">\n";
                        echo "<form method=\"post\" action=\"$PATH_INFO\">\n";
                        $sql="select * from $table_user where admin!=1";
                        $result=mysql_query($sql,$link);
                        echo "<tr>\n";
                        echo "<td bgcolor=\"#ccffcc\">id</td><td bgcolor=\"#ccffcc\">用户名</td><td
bgcolor=\"#ccffcc\">昵称</td><td bgcolor=\"#ccffcc\">用户介绍</td><td bgcolor=\"#ccffcc\">操作类型</td>\n";
                        echo "</tr>\n";
                        while($rows=mysql_fetch_array($result))
                        {
                                echo "<tr>";
                                echo "<input type=hidden name=id[$i] value=".$rows[id].">";
                                echo "<td bgcolor=\"#eeeeff\">".$rows[id]."</td>";
                                echo "<td bgcolor=\"#eeeeff\">".$rows[username]."</td>";
                                echo "<td bgcolor=\"#eeeeff\">";
                                echo "<input type=text name=nickname[$i] value=\"".$rows[nickname]."\" size=6>";
                                echo "</td>";
                                echo "<td bgcolor=\"#eeeeff\"><textarea name=description[$i] rows=3
cols=20>".$rows[description]."</textarea></td>";
```

```
                    echo "<td bgcolor=\"#eeeeff\"><input type=radio name=action[$i] value=del checked=1>删
除<br><input type=radio name=action[$i] value=edit>修改<br></td>";
                    echo "</tr>";
                    $i++;
                }                                       //循环显示用户信息
                echo "<tr>";
                echo "<td colspan=\"5\" bgcolor=\"#eeffee\"><center>";
                echo "<input type=submit value=\"确认提交\">";
                echo "</center></td>";
                echo "</form>\n";
                echo "</table>\n";
                echo "</center>\n";
                echo "</body>\n";
                echo "<html>\n";
            }
            else                                        //如果表单变量已经发送
            {
                require "19-1.php";
                $link=mysql_connect($host,$user,$pass);
                mysql_select_db($db_name,$link);
                for($i=0;$i<count($_POST[action]);$i++)
                {
                    $temp1=$_POST[id][$i];              //循环获取变量
                    $temp2=$_POST[nickname][$i];
                    $temp3=$_POST[description][$i];
                    if($_POST[action][$i]=="del")
                    {
                        $sql="delete from $table_user where id='$temp1'";
                    }
                    else
                    {
                        $sql="update $table_user set nickname='$temp2',description='$temp3'";
                    }
                    mysql_query($sql);                  //执行相关 SQL 操作
                }
                echo "<html>\n";
                echo "<head>\n";
                echo "<title>博客程序</title>\n";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=19-15.php\">";
                echo "</head>\n";
                echo "<body>\n";
                echo "处理成功，正在返回";
                echo "</body>";
                echo "</html>";
            }
        }
        else                                            //如果用户不是管理员
        {
            echo "普通用户没有该权限!";
```

```
        }
}
else                                                    //如果用户没有登录
{
        echo "你还没有登录,点<a href=19-8.php>这里</a>进行登录!";
}
?>
```

将以上代码保存为“19-15.php”，以备后用。单击图 19.14 左侧的“用户管理”链接，其执行结果如图 19.15 所示。

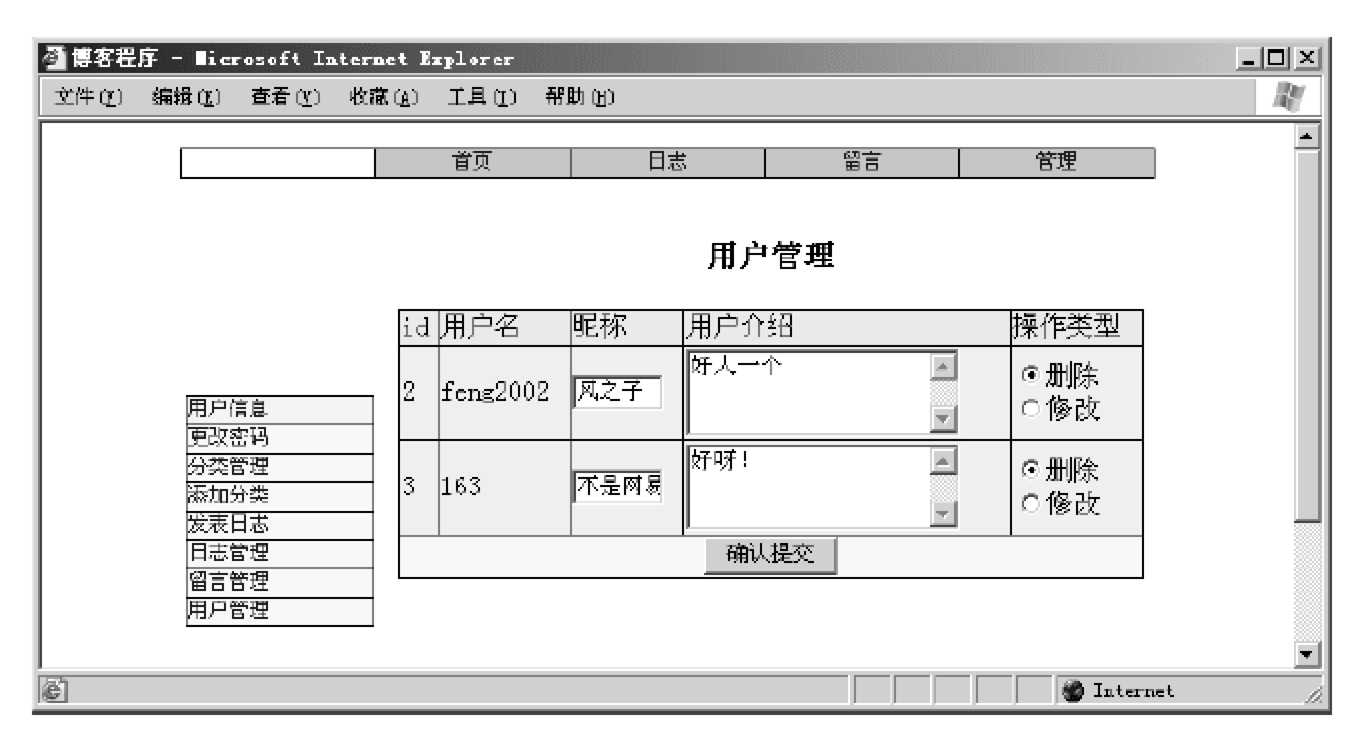


图 19.15　用户管理执行结果

从图 19.15 可以看出，管理的对象是除了管理员之外的用户。不过一般情况下，就算是管理员也不能随意更改用户的昵称和用户介绍。更不能把用户从列表中删除。但是，出于研究目的，这些知识还是应该掌握。具体操作方法与修改类别相类似，不再赘述。

19.5.9　添加新类别页面

博客程序在安装过程中为用户建立了一个默认的日志类别。这显然不能满足用户的需要。如一个程序爱好者，可能会写 PHP、ASP 或者 JSP 等多种类型的日志；一个游戏爱好者可能会写网络游戏、单机游戏、家用机游戏等多方面的日志。所以要给予用户创建日志的权限。实现该功能并不复杂，在类别的表中添加一项记录，然后在用户签写新的日志时，选择相应的类别名称即可。

具体内容请看如下代码：

```
<?
if($_COOKIE[username])                                  //如果用户已经登录
{
        require "19-1.php";
        $link=mysql_connect($host,$user,$pass);
        mysql_select_db($db_name,$link);
        $sql="select * from $table_user where username='$_COOKIE[username]'";
        $result=mysql_query($sql,$link);
        $row=mysql_fetch_array($result);
        if ($row[admin]=="1")                           //验证用户如果为管理员
```

```
    {
        if(!$_POST[sortname])                           //显示 HTML
        {
            echo "<script language=\"javascript\">\n";
            echo "function juge(theForm)\n";
            echo "{\n";
            echo "\tif (theForm.sortname.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入类别名称！ \");\n";
            echo "\t\ttheForm.sortname.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
            echo "\tif (theForm.description.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入类别描述！ \");\n";
            echo "\t\ttheForm.description.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
            echo "}\n";
            echo "</script>\n";
            echo "<center>\n";
            echo    "<table    cellpadding=\"1\"    cellspacing=\"1\"    align=\"center\"    bgcolor=\"#000000\"
width=\"100%\">\n";
            echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
            echo "<tr bgcolor=\"#cccc99\">\n";
            echo "<td colspan=\"2\" align=\"center\">添加新类别</td>\n";
            echo "</tr>\n";
            echo "<tr    bgcolor=\"#cccc99\">\n";
            echo "<td>类别名称</td>\n";
            echo "<td><input type=\"text\" name=\"sortname\"></td>\n";
            echo "</tr>\n";
            echo "<tr bgcolor=\"#cccc99\">\n";
            echo "<td>类别描述：</td>\n";
            echo "<td><textarea rows=\"10\" cols=\"40\"   name=\"description\"></textarea></td>\n";
            echo "</tr>\n";
            echo "<tr bgcolor=\"#cccc99\">\n";
            echo "<td colspan=\"2\"><center><input type=\"submit\" value=\"确认提交\"></center></td>\n";
            echo "</tr>\n";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "<html>\n";
        }
        else                                            //执行操作
        {
            $sortname=$_POST[sortname];
            $description=$_POST[description];
            require "19-1.php";
            $link=mysql_connect($host,$user,$pass);
```

```
                mysql_select_db($db_name,$link);
                $sql="insert into $table_sort(sortname,description)values('$sortname','$description')";
                mysql_query($sql,$link);                    //插入新类别
                echo "<html>\n";
                echo "<head>\n";
                echo "<title>博客程序</title>\n";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=19-16.php\">";
                echo "</head>\n";
                echo "<body>\n";
                echo "添加类别成功，正在返回";
                echo "</body>";
                echo "</html>";
            }
        }
        else
        {
            echo "普通用户没有该权限!";
        }
    }
    else
    {
        echo "你还没有登录,点<a href=19-8.php>这里</a>进行登录!";
    }
    ?>
```

将以上代码保存名称为“19-16.php”。这样，当有管理员登录系统后，就可以调用该页面来创建新的日志类别了。执行该 PHP 文件，其执行结果如图 19.16 所示。

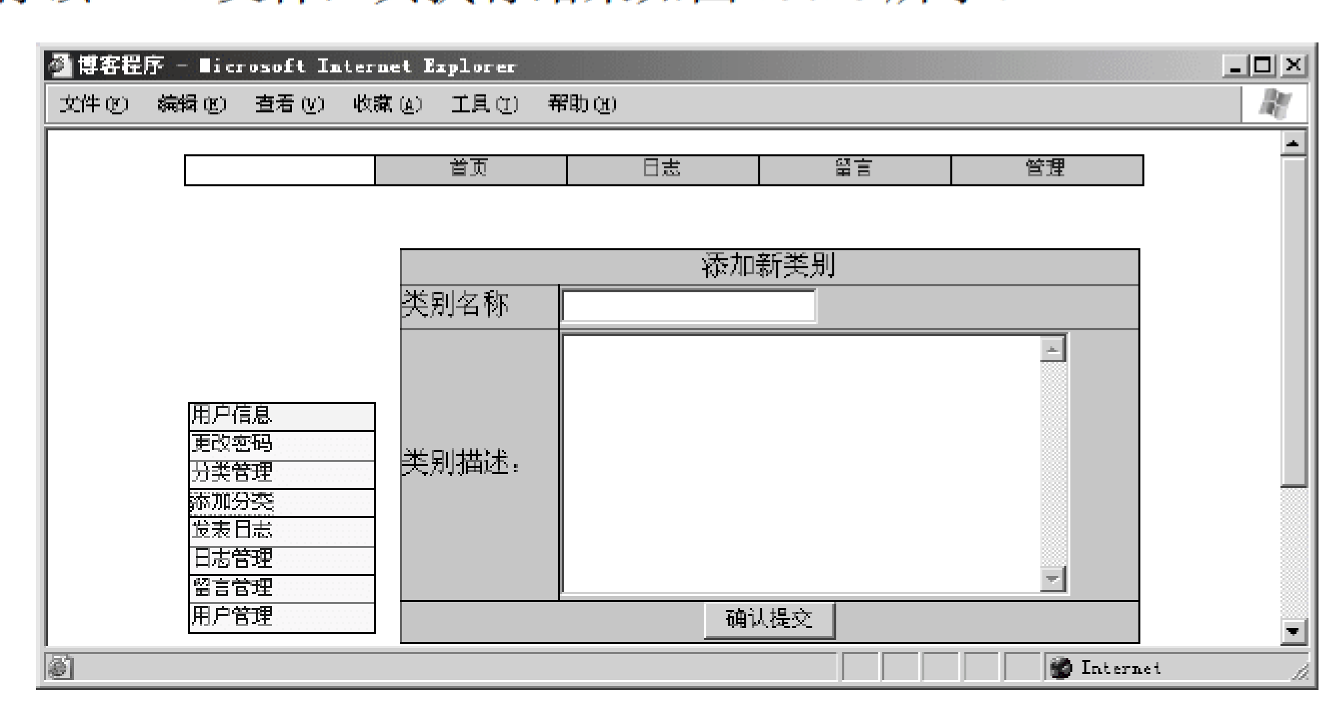


图 19.16　添加分类执行结果

这里只需要输入新的类别名称及类别描述，然后单击“确认提交”按钮即可。执行完操作之后重新跳回到重新输入新类别的页面，不再重复。当时并不会出现结果。不过，在执行操作“分类管理”时或者返回到首页就能看到结果。

19.5.10　留言显示与发表页面

一个简单的留言簿，可以有效的在管理者与普通浏览者之间连起一条纽带。所以通常的博客程序

都会附带一个留言簿。这里介绍的博客的程序也有一个简单的留言簿。由于只是一个附带的产品，所以功能都尽可能的简单，并把显示与用户添加的功能集成到一个文件之中。

具体内容参看以下代码：

```
<?
require "19-3.php";                              //调用头文件
if(!$_POST[title])                               //如果没有发送表单变量，则显示 HTML
{
echo "<table width=\"80%\">\n";
echo "<tr>";
echo "<td width=\"20%\">";
require "19-4.php";
echo "</td>";
echo "<td width=\"80%\">";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" align=\"center\" bgcolor=\"#000000\">\n";
require "19-1.php";
$link=mysql_connect($host,$user,$pass);
mysql_select_db($db_name,$link);
$sql="select id from $table_gbook";
$result=mysql_query($sql,$link);
$msg_count=mysql_num_rows($result);//总条数
$p_count=ceil($msg_count/10);//总页数
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">全部留言</td>";
echo "</tr>";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";
if ($_GET["page"]==0 && !$_GET["page"])
$page=1;
else
$page=$_GET[page];
$s=($page-1)*10+1;
$s=$s-1;
$sql="select * from $table_gbook order by id desc limit $s, 10";
$result=mysql_query($sql,$link);
$nums=mysql_num_rows($result);
if($nums<1) echo "还没有任何留言记录！ ";
else
{
	while($rows=mysql_fetch_array($result))
	{
		echo $rows[title]."</a>";
		echo "<p>";
		echo $rows[author]."于".$rows[date]."留言：";
		echo "<p>";
		echo $rows[content];
		echo "<hr width=100%>";
	}
}
```

```
echo "</td>";
echo "</tr>";
echo "<tr>";
echo "<td bgcolor=\"#eeeeff\">";
$prev_page=$page-1;
$next_page=$page+1;
if ($page<=1){
      echo "第一页 | ";
}
else{
      echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
}
if ($prev_page<1){
echo "上一页 | ";
}
else{
      echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
}
if ($next_page>$p_count){
      echo "下一页 | ";
}
else{
      echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
}
if ($page>=$p_count){
      echo "最后一页</p>\n";
}
else{
      echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
}
echo "</td>";
echo "</tr>";
echo "</table>";
echo "<p>";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" align=\"center\" bgcolor=\"#000000\">\n";
echo "<form action=19-17.php method=post>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "发表新留言：";
echo "</td></tr>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "留言题目：<input type=text name=title>";
echo "</td></tr>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "留言内容：<br><textarea name=content rows=5 cols=40></textarea>";
echo "</td></tr>";
echo "<tr bgcolor=\"#eeffee\"><td>";
echo "<input type=submit value=\"发表\">";
echo "</td></tr>";
echo "</form>";
```

```
echo "</table>";
echo "</td>";
echo "</tr>";
echo "</table>";
echo "</td>";
echo "</tr>";
echo "</table>";
}
else                                                    //如果已经发送表单变量执行操作
{
$title=$_POST[title];
$content=$_POST[content];
$date=$date=date("Y 年 n 月 d 日");
if(!$_COOKIE["username"])
{
      $username="匿名";
}
else
{
      $username=$_COOKIE["username"];
}
require "19-1.php";
$link=mysql_connect($host,$user,$pass);
mysql_select_db($db_name,$link);
$sql="insert into $table_gbook(author,title,content,date)values('$username','$title','$content','$date')";
mysql_query($sql,$link);                                //插入留言内容到表中
echo "<meta http-equiv=\"refresh\" content=\"2; url=19-17.php\">";
echo "</head>\n";
echo "<body>\n";
echo "添加留言成功，正在返回";
echo "</body>";
echo "</html>";
}
?>
```

把以上代码保存为“19-17.php”，可方便普通用户甚至是匿名用户发表、浏览留言。单击主面板上方的“留言”链接，相当于执行 19-17.php。其执行结果如图 19.17 所示。此时没有任何留言，可以使用下方给出的表单添加新的留言。输入留言标题与留言内容之后，单击“发表”按钮即可。执行结果如图 19.18 所示。

从图 19.18 可以看出，正确显示出了用户所提交的留言内容。说明程序正常运行。

19.5.11　新用户注册页面

一个博客程序的用户当然不能只有管理员，也应该有更多的普通使用者。这一小节来创建新用户注册页面。在进行注册时要先判断用户名是否存在，如果存在给出相应的提示；反之，则将用户输入信息添加到表中。

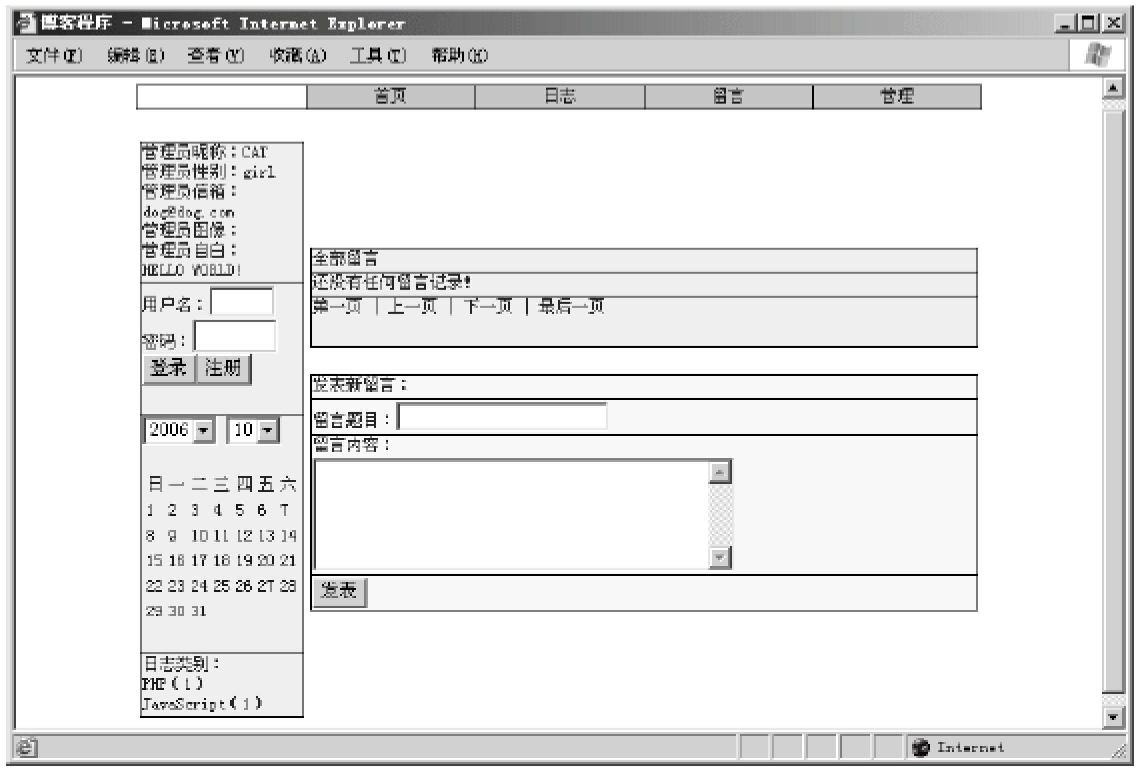


图 19.17　查看留言页面执行结果



图 19.18　发表留言执行结果

具体内容参看如下代码：

```
<?
if(!$_POST[admin])                              //如果没有默认参数，则显示 HTML
{
	echo "<html>\n";
	echo "<head>\n";
	echo "<title>注册新用户</title>\n";
	echo "</head>\n";
	echo "<body>\n";
	echo "<script language=\"javascript\">\n";
	echo "function juge(theForm)\n";
	echo "{\n";
	echo "\tif (theForm.admin.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入用户名称！ \");\n";
	echo "\t\ttheForm.admin.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.pass.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入用户密码！ \");\n";
	echo "\t\ttheForm.pass.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.pass.value.length < 8 )\n";
	echo "\t{\n";
	echo "\t\talert(\"密码至少要 8 位！ \");\n";
	echo "\t\ttheForm.pass.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.re_pass.value !=theForm.pass.value)\n";
	echo "\t{\n";
	echo "\t\talert(\"确认密码与密码不一致！ \");\n";
```

```
echo "\t\ttheForm.re_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.nickname.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入昵称！ \");\n";
echo "\t\ttheForm.nickname.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">\n";
echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">注册新用户</font></td>\n";
echo "</tr>\n";
echo "<tr   bgcolor=\"#cccc99\">\n";
echo "<td>用户名：（后台登录）</td>\n";
echo "<td><input type=\"text\" name=\"admin\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>用户密码：（不小于 8 位）</td>\n";
echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>确认密码：</td>\n";
echo "<td><input type=\"password\" name=\"re_pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>用户 E-mail：（可选）</td>\n";
echo "<td><input type=\"text\" name=\"email\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>用户昵称：（前台显示）</td>\n";
echo "<td><input type=\"text\" name=\"nickname\"></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td>用户介绍：</td>\n";
echo "<td><textarea rows=\"5\" cols=\"30\" name=\"description\"></textarea></td>\n";
echo "</tr>\n";
echo "<tr bgcolor=\"#cccc99\">\n";
echo "<td colspan=\"2\"><center><input type=\"submit\" value=\"下一步\"></center></td>\n";
echo "</tr>\n";
echo "</form>\n";
echo "</table>\n";
echo "</center>\n";
echo "</body>\n";
echo "<html>\n";
```

```
}
else                                        //如果有 POST 参数，则执行操作
{
    $username=$_POST[admin];                //获得参数
    $password=md5($_POST[pass]);
    $nickname=$_POST[nickname];
    $email=$_POST[email];
    $description=$_POST[description];
    require "19-1.php";
    $link=mysql_connect("$host","$user","$pass") or die(mysql_error());
    mysql_select_db($db_name,$link);        //选择数据库
    $sql="select username from $table_user where username='$username'";
    $result=mysql_query($sql,$link);
    $nums=mysql_num_rows($result);          //获取重名用户
    if($nums!=0) echo "用户名已经存在!点<a href='#' onclick=history.go(-1)>这里</a>返回";
    else                                    //如果不存在重名用户
    {
        $sql="insert                                                          into
$table_user(username,password,nickname,email,description)values('$username','$password','$nickname','$em
ail','$description')";
        mysql_query($sql,$link) or die(mysql_error());  //发送添加用户信息的 SQL 请求
        echo "<html>\n";
        echo "<head>\n";
        echo "<title>注册新用户</title>\n";
        echo "</head>\n";
        echo "<body>\n";
        echo "</center>";
        echo    "<table    width=\"80%\"    cellpadding=\"1\"    cellspacing=\"1\"    align=\"center\"
bgcolor=\"#000000\">\n";
        echo "<tr bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"5px\">注册用户</font></td>\n";
        echo "</tr>\n";
        echo "<tr   bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"3px\">成功注册！ </font></td>\n";
        echo "</tr>\n";
        echo "<tr   bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\">点<a href=\"19-8.php\">这里</a>登录</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        echo "</center>";
        echo "</body>\n";
        echo "</html>\n";
    }
}
?>
```

把以上所示代码保存为“19-18.php”。在 PHP 执行环境中运行该 PHP 文件，其执行结果如图 19.19 所示。按要求输入想要注册的用户名，再输入相应的密码、信箱、简介等内容即可。单击“下一步”按钮，将出现如图 19.20 所示的执行结果。

图 19.19　新用户注册页面执行结果



图 19.20　成功注册新用户执行结果

出现如图 19.20 所示画面，说明新用户注册成功。

19.5.12　用户退出登录页面

通常不需要专门创建这样一个文件，因为这里采用的 Cookie 机制是最短生命期，即浏览器关闭，登录用户注册的 Cookie 即失效。但是为了方便用户操作还是有必要创建这样一个页面。实现功能相当简单，清除用户注册 Cookie 返回首页。

具体内容请参看以下代码：

```
<?
setcookie("username","");                                    //把相关变量置为空值
echo "<html>";
echo "<head>";
echo "<title>退出登录</title>";
echo "<meta http-equiv=\"refresh\" content=\"2; url=19-3.php\">";
echo "</head>";
echo "<body>";
echo "已经退出，两秒后返回！";
echo "</body>";
?>
```

保存以上代码为“19-19.php”。当登录用户单击图 19.7 左侧中间的“退出”链接相当于执行 19-19.php。程序将会清空用户登录的 Cookie 完成退出任务。

至此，整个博客系统的所有页面均创建完毕。

19.6　进一步完善

虽然这个简单的博客程序已经能够正常运行了。但还存在很多不足，与网上同类程序相比功能上还差得比较远。在这里总结本程序的不足，并给出相关的解决思路，有兴趣的读者可以自行解决。比

较来说，有以下几个方面。

没有对用户输入内容进行检测。用户的输入内容五花八门，更有不怀好意的用户甚至会输入恶意代码。解决这个问题可以通过相应的字符串操作函数，对用户输入信息进行处理。

没有文件上传功能。有时用户需要通过博客程序把本地的文件，如图片上传到服务器。解决这个问题可以为管理员专门再建一个文件上传的页面。关于如何上传文件请参看本书第 8 章 PHP 文件应用。

没有友情链接。通常的博客程序都有一个友情链接，以使管理者可以与同类的，或者其他朋友的网站建立链接，便于交流。这个实现起来也不复杂，只需要建一个专门的表用于放置友情链接信息，然后在显示时显示相应的网站链接即可。

经过功能的增加，该博客程序的功能相对来说就比较完善了。

19.7　关于 RSS 内容聚合

现在所见到博客站点中通常都有一个 RSS 的链接，这也是博客程序一个明显的特征。网络上曾经流行这样一段话“blogging without RSS is like swimming without water”没有了 RSS 的写博客就相当没有水的游泳。看起来 RSS 对博客起着相当重要的作用。虽然这已经不属于 PHP 编程的范畴，但作为相关知识，读者有必要作一下了解。

那么究竟什么是 RSS，它与博客程序又有什么联系呢？

RSS（Really Simple Syndication）是一种描述和同步网站内容的格式，是目前使用最广泛的 XML 应用。RSS 搭建了信息迅速传播的一个技术平台，使得每个人都成为潜在的信息提供者。发布一个 RSS 文件后，这个 RSS Feed 中包含的信息就能直接被其他站点调用，而且由于这些数据都是标准的 XML 格式，所以也能在其他的终端和服务中使用。

如果从 RSS 阅读者的角度来看，完全不必考虑它到底是什么意思，只要简单地理解为一种方便的信息获取工具就可以了。RSS 获取信息的模式与加入邮件列表（如电子杂志和新闻邮件）获取信息有一定的相似之处，也就是可以不必登录各个提供信息的网站而通过客户端浏览方式（称为“RSS 阅读器”）或者在线 RSS 阅读方式这些内容。例如，通过一个 RSS 阅读器，可以同时浏览新浪新闻，也可以浏览搜狐或者百度的新闻（如果用户采用了 RSS 订阅的话）。

在许多新闻信息服务类网站，会看到这样的按钮 RSS XML，有的网站使用一个图标，有的同时使用两个，这就是典型的提供 RSS 订阅的标志，这个图标一般链接到订阅 RSS 信息源的 URL。当然，即使不用这样的图标也是可以的，只要提供订阅 RSS 信息源的 URL 即可。

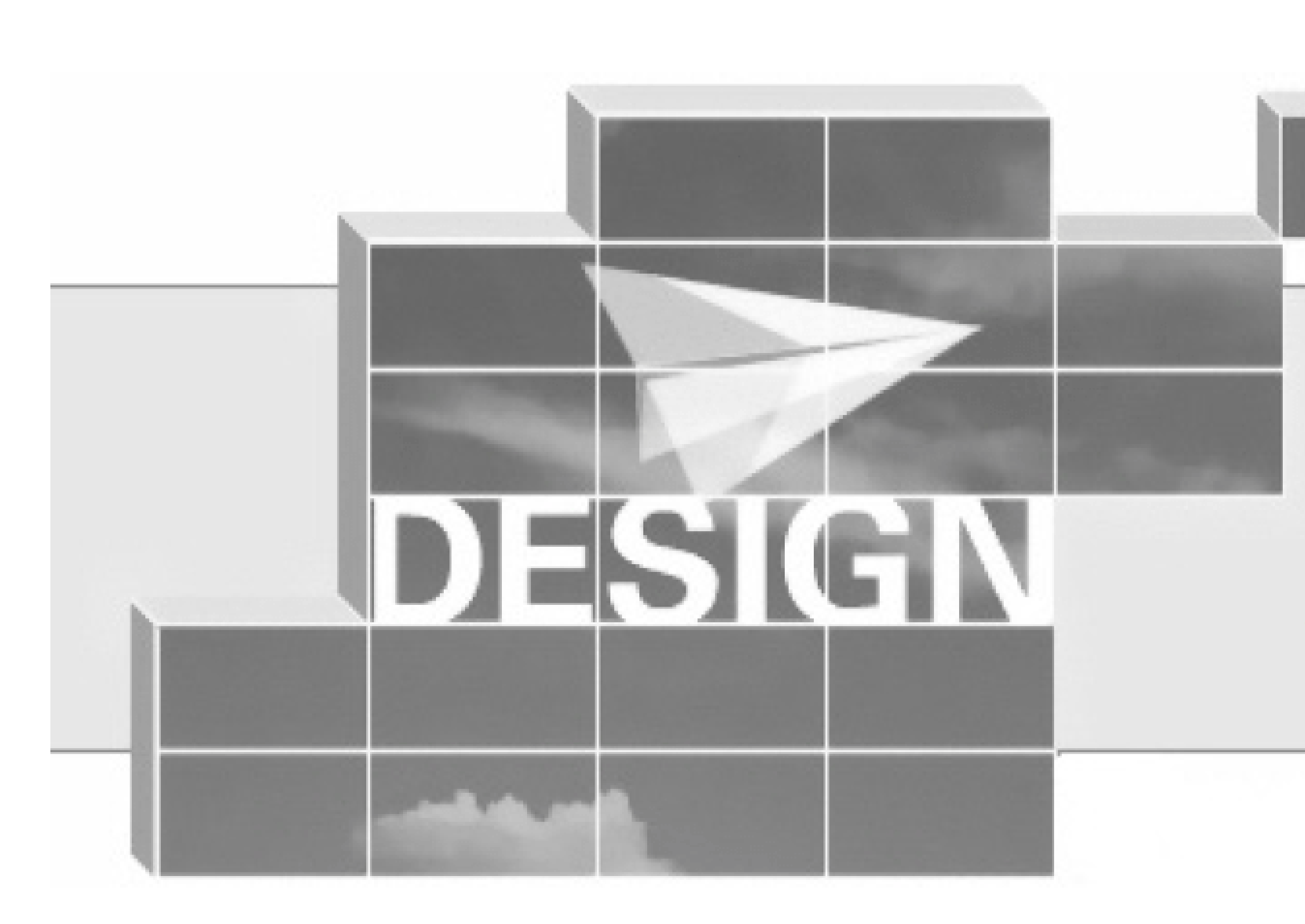
归纳以上说法，概括来说，RSS 其实就是一种站点与站点之间的信息共享方式（也称内容聚合）。使用该方式之后，所有的使用者都变成了潜在的信息提供者，这样会更有利于普通网络使用者发现网站内容的更新。

关于 RSS 的知识就简要介绍的这里。有兴趣的读者可以去查阅相关的专门书籍以了解更多内容。

19.8　本 章 小 结

这一章为读者介绍了什么是 BLOG，普及了一下 BLOG 的知识。而重点是介绍如何用 PHP 程序来完成一个简单的 BLOG 程序。把一个简单的博客程序分解为 19 个组成部分，每部分，逐个来讲解每个模块的功能及代码实现。并对执行结果进行了调试分析。通过本章学习，读者对什么是博客、如何实现一个博客都会有一个更深层次的认识。同时通过本章的学习，对于用 PHP 进行 MySQL 数据库的操作更会有新的提高。

第 4 篇



PHP 综合应用实例

第20章　简单的BBS系统

第21章　网上商城全站系统

第20章

简单的 BBS 系统

- 什么是论坛
- 安装文件制作
- 用户注册与登录
- 论坛、分论坛显示
- 主题显示与回复
- 后台管理
- 防掘墓处理

BBS 是英文 Bulletin Board System 的缩写，翻译成中文为“电子布告栏系统”或“电子公告牌系统”。BBS 是一种电子信息服务系统，它向用户提供了一块公共电子白板，每个用户都可以在上面发布信息或提出看法，早期的 BBS 由教育机构或研究机构管理，现在多数网站上都建立了自己的 BBS 系统，供网民通过网络来结交更多的朋友，表达更多的想法。目前国内的 BBS 已经十分普遍。这一章就通过 PHP 来设计一个简单的 BBS 系统。本章内容包括 BBS 相关数据表的设计、用户的注册与登录设计、用户发贴设计、主题的回复与显示设计、管理员对贴子的管理、用户的管理等内容。通过本章的学习，读者可以更进一步提高用 PHP 解决实际问题的能力。

BBS 大致可以分为 5 类：

- 校园 BBS：CERNET 建立以来，校园 BBS 很快地发展了起来，目前很多大学都有了 BBS，几乎遍及全国上下。像清华大学、北京大学等都建立了自己的 BBS 系统，清华大学的 BBS 水木清华很受学生和网民们的喜爱。大多数 BBS 是由各校的网络中心建立的，也有私人性质的 BBS。
- 商业 BBS 站：这里主要是进行有关商业的商业宣传，产品推荐等等，目前手机的商业站、计算机的商业站、房地产的商业站比比皆是。
- 专业 BBS 站：这里所说的专业 BBS 是指部委和公司的 BBS，它主要用于建立地域性的文件传输和信息发布系统。
- 情感 BBS：主要用于交流情感，是许多娱乐网站的首选。
- 个人 BBS：有些个人主页的制作者们在自己的个人主页上建设了 BBS，用于接受别人的想法，更有利于与好友进行沟通。

这一章就通过 PHP 来设计一个简单的 BBS 系统。本章内容包括：BBS 相关数据表的设计、用户的注册与登录设计、用户发贴设计、主题的回复与显示设计、管理员对贴子的管理、用户的管理等内容。通过本章的学习，使读者更进一步掌握用 PHP 解决实际问题的能力。

20.1　设计数据库表

数据库表的设计是 Web 应用程序开发中一个很重要的环节，表结构设计的好坏直接影响着应用程序的执行效率。一个设计清晰、结构合理的表是高效应用程序的基础。这一节就来分析一下，简单 BBS 系统的表结构应该如何设计。其中包括 3 个部分：用户数据表的设计、论坛分类表的设计、贴子数据表的设计。

20.1.1　用户数据表的设计

这一小节先来学习简单的 BBS 用户数据表的设计。

一般的用户表应该包括如下内容：索引 ID、用户名（用户登录时使用的名称）、密码（用户登录时所使用的密码）、昵称（与用户名不同，用户名在用户登录时显示，其他时候不显示，用户名具有唯一性；而昵称则在用户发贴时在论坛上显示，所有用户昵称允许出现重复）、性别（记录用户的性别）、电子信箱（判断用户身份的一个重要标志）、论坛图像（用户发贴时显示的图像）、论坛签名（跟在发的贴子后面的签名）、发贴数（记录用户总共的发贴数量）、注册时间（记录用户注册的时间）、版主（记录用户的权限，分为普通用户、版主、超级版主等几种）、等级（由发贴数所规定的一个等级，发贴数越多，等级越高）、其他（备用，以放置用户其他信息）。每个字段及类型如表 20.1 所示。

表 20.1　用户数据表 members 字段类型及含义

字 段 名	类　型	长　度	作　用	其 他 属 性
id	int	5	记录每个用户的编号，具有唯一性	auto_increment primary key
name	varchar	12	记录每个用户的用户名	

续表

字 段 名	类　　型	长　　度	作　　用	其 他 属 性
password	varchar	40	记录用户的密码，用MD5加密保存	
nickname	varchar	12	用户昵称	
sex	enum('boy','girl')		记录用户性别	default 'boy'
email	varchar	80	记录用户电子信箱	
photo	varchar	80	记录用户图像	
q_name	varchar	200	留言时显示的个性签名	
post_num	int	5	记录用户发贴数	
reg_date	varchar	20	记录用户注册时间	
admin	int	1	记录用户的类型（管理员或者普通用户）	
levle	int	5	记录用户等级	
other	varchar	200	备用项	

20.1.2　论坛分类数据表的设计

一个论坛，通常都包括很多子栏目。如一个以编程为主题的论坛下面可能分为技术区与娱乐区，技术区可能又会分为网络编程、应用程序编程等，下面可能会有更为复杂的划分。娱乐区也可能分为灌水区、贴图区等。所有这些分类信息都要保存在一个专门的表之中。

按照以上考虑，论坛分类数据表应包括以下内容：索引 ID、主分类 ID（论坛的大类，如果某一分类项为主分类，则该项值为 0，反之该项值为其主分类 ID 的 ID 号）、论坛分类名称（记录某一个分类的名称）、论坛分类介绍（向用户介绍该分类的主要内容）、最后贴子的 ID 号（记录最后一条贴子内容以显示给用户）、该分类的总帖子数（记录所属该分类的所有贴子的数量）、该分类的总主题数（贴子数包括主题数）、其他（备用，以存放该分类的其他信息）。每个字段及其类型如表 20.2 所示。

表 20.2　论坛分类数据表 topic 的字段类型及含义

字 段 名	类　　型	长　　度	作　　用	其 他 属 性
id	int	5	记录每个类别的编号，具有唯一性	auto_increment primary key
p_id	int	5	记录该类别所属主类别，如果该类别即为主类别，则该值为0	
topic_name	varchar	12	记录该类别的名称	
topic_description	varchar	80	记录该类别的简介	
last_post_id	int	5	记录该类别最后贴号	
post_count	int	5	记录该类别总贴数	
post_m_count	int	5	记录该类别主题数	
other	varchar	200	其他	

至此，论坛分类表的设计也告一段落。

20.1.3　贴子数据表的设计

简单的论坛系统的贴子表采用这样的结构：索引 ID、所属分类 ID（记录该贴子属于哪一个论坛分类，该值等于论坛分类表中的 ID 号）、回复 ID（该贴子是对哪一条贴子的回复，如果贴子本身就是主题，则该值为 0）、贴子作者的 ID 号（这一条记录贴子作者的索引号，以便通过该 ID 号显示作者的相关信息）、贴子作者的名称、贴子作者的 IP 地址（记录作者的 IP 地址）、贴子类型（可分为求助贴、原创贴、转贴等多个类型）、贴子的标题、贴子的内容（这一条最为关键，记录所发贴子的内容）、贴子发送时间（记录贴子发送于何时）、回复时间（记录该贴子最后回复的时间）、其他（备用，记录贴子的其他信息）。

每个字段及其类型如表 20.3 所示。

表 20.3　贴子记录表 posts 的字段类型及含义

字　段　名	类　　型	长　　度	作　　用	其 他 属 性
id	int	5	记录每个贴子的编号，具有唯一性	auto_increment primary key
topic_id	int	5	记录该贴子所属的论坛类别	
re_id	int	5	记录该贴子所属的主题。如果该贴子即为主题，则该值为0	
poster_id	int	5	记录发贴者的编号	
poster_ip	varchar	23	记录发贴者的IP地址	
poster	varchar	12	记录发贴者的名字	
poster_type	varchar	10	记录贴子类型	
title	varchar	40	记录贴子标题	
content	text		记录贴子内容	
view_count	int	5	记录贴子浏览量	
re_count	int	5	记录贴子回复量	
post_time	varchar	40	记录发贴时间	
post_re_time	varchar	40	记录最后回复时间	
other	varchar	200	备用	

贴子数据表的设计也宣告完成。至此论坛所用到的 3 个表都已经建立完成。20.2 节就开始通过代码来逐步实现论坛的所有功能。

20.2　准 备 工 作

从这一节开始将通过具体代码来一步步实现论坛的所有功能。不过在具体实施之前，有一些准备工作必须要先完成。这一节就来解决这个问题。

20.2.1　配置文件的创建

通常情况下，程序所访问的主机名、连接主机的用户名、用户密码及数据库名都是固定不变的，要创建的各种表名，在创建后也是不变的，所以把这些重要的变量单独拿出做成配置文件是很有必要的。该配置文件中存放着程序所需要的数据库的主机名、连接主机的用户名、密码、数据库名、表名等各项信息。

下面就来建立配置文件。具体内容参看以下代码：

```
<?
$db_host="localhost";                                    //主机名
$db_user="root";                                         //用户名
$db_pass="";                                             //密码
$db_name="test";                                         //数据库名
$table_members="members";                                //用户表
$table_topic="topic";                                    //分类表
$table_posts="posts";                                    //贴子表
$link=mysql_connect($db_host,$db_user,$db_pass);         //连接主机
mysql_select_db($db_name,$link);                         //选择数据库
?>
```

把以上代码保存为“20-1.php”，以备后用。

20.2.2　安装文件的创建

配置文件创建完成，这一小节开始来创建安装文件。该文件的作用是先给用户一个界面，让用户完成输入相关内容。用户输入完毕后，在后台执行以下操作：分别创建用户表、分类表、贴子表；为用户表添加用户输入的管理员信息、为分类表添加默认分类、为贴子表添加默认的贴子；给出用户相应的信息。

具体内容请参看如下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>安装程序</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<style>\n";
echo "* {
padding: 0;
margin: 0;
}
body {
    font-family: verdana, sans-serif;
    font-size: 10pt;
```

```
        background-color: #FFFFEE;
        padding: 25px 0px 25px 0px;
}
a:link, a:active, a:visited {
        color: #336699;
        text-decoration: underline;
}
a:hover {
        color: #7F0000 !important;
        text-decoration: none;
}
select option {
        padding-right: 3px;
}
#content {
        padding: 0px 25px 10px 25px;
}
p, table, pre, h2, h3, ul, ol, dl {
        margin: 0px 0px 15px 0px;
}
p.important {
        background-color: #EFDFBF;
        padding: 10px;
        font-size: 8pt;
}
p#submit, p#submit input {
        text-align: center;
        font-weight: bold;
}
p#submit input {
        padding: 5px;
}
h2 {
        color: #336699;
        font-weight: normal;
        font-size: 14pt;
        border-bottom: 1px solid silver;
}
h3 {
        color: #333;
        font-weight: bold;
        font-size: 10pt;
}
ul, ol {
        margin-left: 35px;
}
dl dt {
        font-weight: bold;
        color: #333;
```

```
}
dl dd {
	margin-left: 35px;
	margin-bottom: 5px;
}
table {
	background-color:#000000;
	border-collapse: collapse;
	margin-left: 0;
	margin-right: 0;
}
table th, table td {
	padding: 5px;
}
td {
	background-color:#cccc99;
}
table th {
	text-align: left;
	color: #336699;
}
table td.title {
	width: 135px;
}";
	echo "</style>\n";
if(!$_POST[admin])                                  //如果没有默认参数，则显示 HTML
{
	echo "<script language=\"javascript\">\n";
	echo "function juge(theForm)\n";
	echo "{\n";
	echo "\tif (theForm.admin.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入管理员名称！\");\n";
	echo "\t\ttheForm.admin.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.pass.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入管理员密码！\");\n";
	echo "\t\ttheForm.pass.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.pass.value.length < 8 )\n";
	echo "\t{\n";
	echo "\t\talert(\"密码至少要 8 位！\");\n";
	echo "\t\ttheForm.pass.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "\tif (theForm.re_pass.value !=theForm.pass.value)\n";
```

```
echo "\t{\n";
echo "\t\talert(\"确认密码与密码不一致！\");\n";
echo "\t\ttheForm.re_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.nickname.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入昵称！\");\n";
echo "\t\ttheForm.nickname.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.pre.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入表前缀！\");\n";
echo "\t\ttheForm.pre.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
echo "<tr>\n";
echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">安装论坛</font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>管理员：（后台登录）</td>\n";
echo "<td><input type=\"text\" name=\"admin\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>管理员密码：（不小于 8 位）</td>\n";
echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>确认密码：</td>\n";
echo "<td><input type=\"password\" name=\"re_pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>管理员 E-mail：（可选）</td>\n";
echo "<td><input type=\"text\" name=\"email\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>管理员昵称：（前台显示）</td>\n";
echo "<td><input type=\"text\" name=\"nickname\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>表的前缀：</td>\n";
echo "<td><input type=\"text\" name=\"pre\" value=\"bbs_\"></td>\n";
echo "</tr>\n";
```

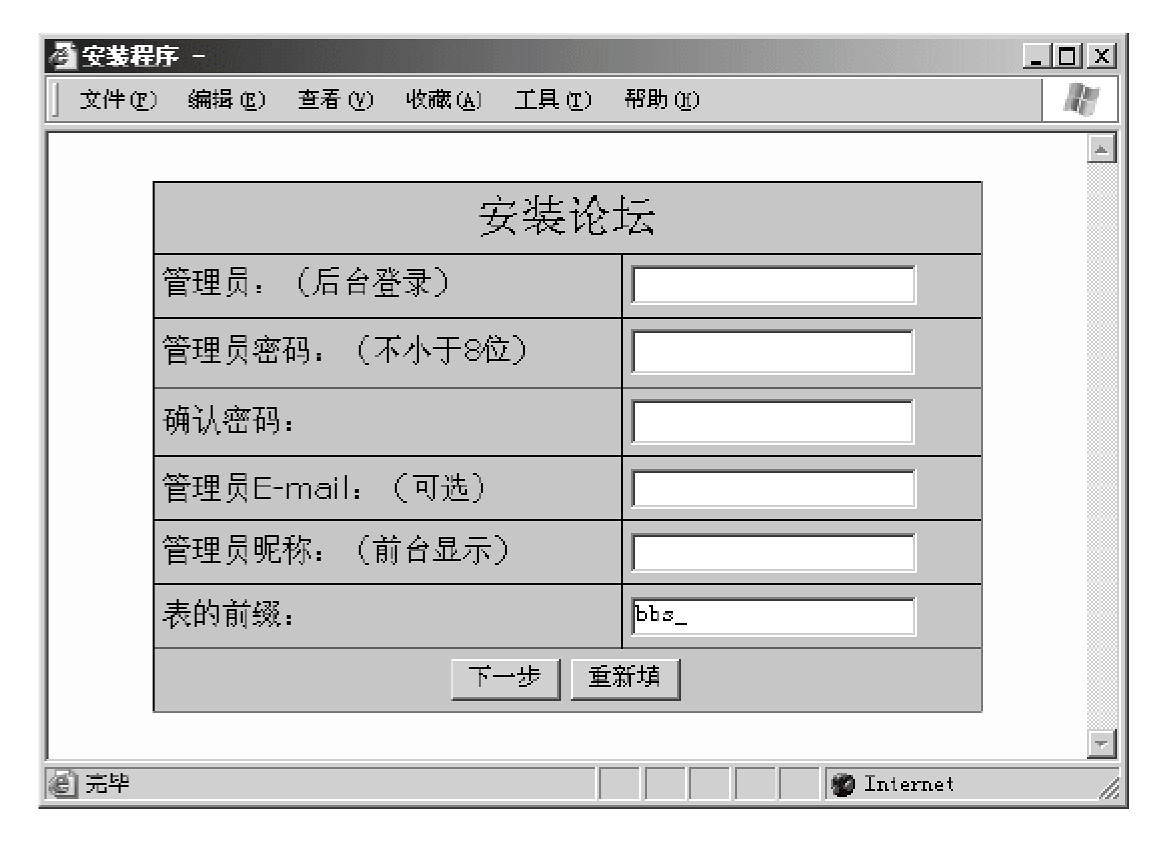
```
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>\n";
        echo "<input type=\"submit\" value=\"下一步\">\n";
        echo "<input type=\"reset\" value=\"重新填\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
}
else                                                //如果有 POST 参数，则执行操作
{
        $name=$_POST[admin];                        //获得参数
        $password=md5($_POST[pass]);                //获得密码，并使用 MD5 进行加密操作
        $nickname=$_POST[nickname];
        $email=$_POST[email];
        $pre=$_POST[pre];
        require "20-1.php";
        $table_members=$pre.$table_members;
        $table_topic=$pre.$table_topic;
        $table_posts=$pre.$table_posts;
        $ip=$_SERVER[REMOTE_ADDR];
        $time=date("Y 年 m 月 d 日");
        $time2=date("G：i：s");
        $sql="create table $table_members(
        id int(5) not null auto_increment primary key,
        name varchar(12) not null,
        password varchar(40) not null,
        nickname varchar(12) not null,
        sex enum('boy','girl') not null default 'boy',
        email varchar(80) not null,
        photo varchar(80) not null,
        q_name varchar(200) not null,
        post_num int(5) not null,
        reg_date varchar(20) not null,
        admin int(1) not null default '0',
        level int(5) not null,
        other varchar(200) not null
        )";
        mysql_query($sql,$link) or die(mysql_error());  //发送创建 member 表的 SQL 请求
        $sql="create table $table_topic(
        id int(5) not null auto_increment primary key,
        p_id int(5) not null,
        topic_name varchar(12) not null,
        topic_description varchar(80) not null,
        last_post_id int (5) not null,
        post_count int(5) not null,
```

```
post_m_count int(5) not null,
other varchar(200) not null
)";
mysql_query($sql,$link) or die(mysql_error());  //发送创建 topic 表的 SQL 请求
$sql="create table $table_posts(
id int(5) not null auto_increment primary key,
topic_id int(5) not null,
re_id int(5) not null,
poster_id int(5) not null,
poster_ip varchar(23) not null,
poster varchar(12) not null,
title varchar(40) not null,
content text not null,
view_count int(5) not null,
re_count int(5) not null,
post_time varchar(40) not null,
post_re_time varchar(40) not null,
other varchar(200) not null
)";
mysql_query($sql,$link) or die(mysql_error());  //发送创建 posts 表的 SQL 请求
$sql="insert into $table_topic(p_id,topic_name,topic_description)values('0','默认主类别 1','系统创建的默认主类别')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加默认主分类的 SQL 请求
$sql="insert into $table_topic(p_id,topic_name,topic_description,last_post_id,post_count,post_m_count)values('1','默认分类别','系统创建的默认分类别','1','1','1')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加默认子分类的 SQL 请求
$sql="insert into $table_members(name,password,nickname,email,post_num,reg_date,admin)values('$name','$password','$nickname','$email','1','$time','3')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加管理员信息的 SQL 请求
$sql="insert into $table_posts(topic_id,poster_id,poster_ip,poster,title,content,post_time,post_re_time)values('2','1','$ip','$name','第一条测试信息','测试发贴是否有效','$time$time2','$time$time2')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加发贴的 SQL 请求
$sql="update $table_topic set post_count='1',post_m_count='1' where id='2'";
mysql_query($sql,$link) or die(mysql_error());  //发送更改分类表的 SQL 请求
$fp=fopen("20-1.php","w+");                     //将更新过的数据写入配置文件
fputs($fp,"<?\n");
fputs($fp,"\$db_host=\"localhost\";\n");
fputs($fp,"\$db_user=\"root\";\n");
fputs($fp,"\$db_pass=\"\";\n");
fputs($fp,"\$db_name=\"test\";\n");
fputs($fp,"\$table_members=\"$table_members\";\n");
fputs($fp,"\$table_topic=\"$table_topic\";\n");
fputs($fp,"\$table_posts=\"$table_posts\";\n");
fputs($fp,"\$link=mysql_connect(\$db_host,\$db_user,\$db_pass);\n");
fputs($fp,"mysql_select_db(\$db_name,\$link);\n");
fputs($fp,"?>");
```

```
        fclose($fp);
        echo "<center>";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\" bgcolor=\"#000000\">\n";
        echo "<tr bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"5px\">安装论坛</font></td>\n";
        echo "</tr>\n";
        echo "<tr   bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"3px\">成功安装！ </font></td>\n";
        echo "</tr>\n";
        echo "<tr bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\"><font size=\"3px\">删除该文件，以减少潜在危险！ </font></td>\n";
        echo "</tr>\n";
        echo "<tr   bgcolor=\"#cccc99\">\n";
        echo "<td align=\"center\">点<a href=\"20-3.php\">这里</a>进入</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        echo "</center>";
        echo "</body>\n";
        echo "</html>\n";
    }
    ?>
```

将上面所显示的代码保存为“20-2.php”。在 PHP 执行环境下运行该 PHP 文件，以完成论坛程序所需用户表的创建。首次运行该 PHP，其执行结果会如图 20.1 所示。

按照要求填入全部信息。其中的表的前缀一项指在建立表时为了避免与已经存在的表出现重名现象为表设置的前缀项，全部填写完毕单击“下一步”按钮，将出现如图 20.2 所示的执行结果。



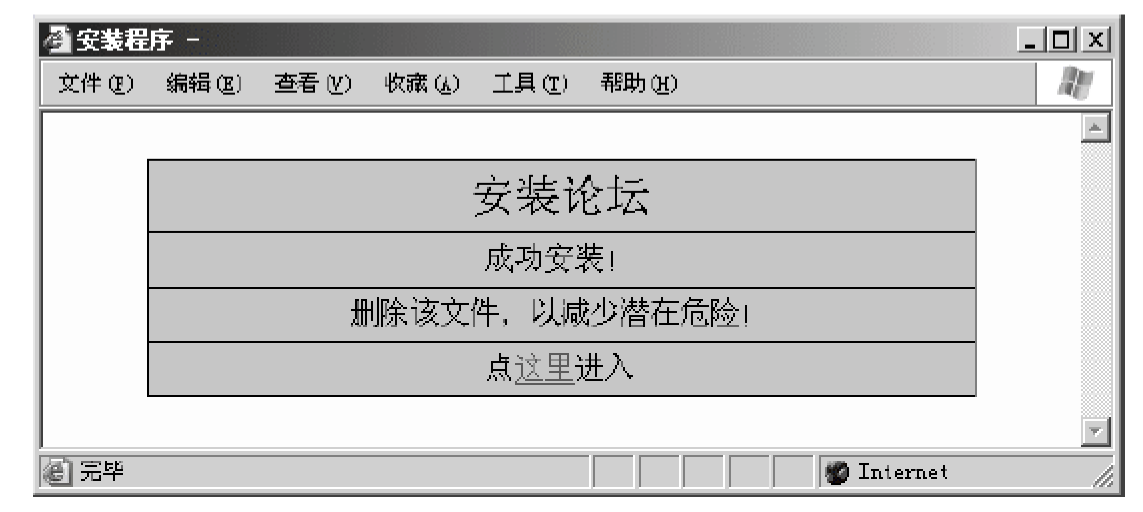



图 20.1　首次运行安装文件的执行结果　　　图 20.2　安装完成执行结果

执行过这一步操作后，系统所需要的 3 个表都已经成功创建完毕，并且还为每个表添加了相应项目。为用户表添加了系统管理员，为分类表添加了默认主分类与子分类，为发贴表添加了第 1 条贴子。

系统安装程序执行过后，准备工作就算是全部完成了。接下来就是每个功能模块的具体实现了。从 20.3 节开始，就通过具体的代码来实现论坛的具体模块。

20.3　用户注册与登录

论坛与用户是密不可分的。用户的多少是评判一个论坛人气的一个重要标准。一个好的论坛往往会说注册用户达到多少万，而一个差点的论坛的注册用户往往很少。既然需要注册用户，就要提供给用户一个友好的注册与登录界面。这一节就来创建用户的注册与登录页。

20.3.1　用户注册页

注册新用户页提供给用户一个表单，以便让用户输入相关内容。其中包括必要的用户名、密码、昵称、性别、电子信箱、选择用户图像、输入发贴签名等。其中的用户名、密码、昵称几项内容为必填项。

具体内容请参考以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>注册新用户</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<style>\n";
echo "* {
padding: 0;
margin: 0;
}
body {
	font-family: verdana, sans-serif;
	font-size: 10pt;
	background-color: #FFFFEE;
	padding: 25px 0px 25px 0px;
}
a:link, a:active, a:visited {
	color: #336699;
	text-decoration: underline;
}
a:hover {
	color: #7F0000 !important;
	text-decoration: none;
}
select option {
	padding-right: 3px;
}
#wrapper {
	width: 650px;
	border: 1px solid silver;
```

```
	margin-left: auto;
	margin-right: auto;
	background-color: #EFEFEF;
	background-image: url(gfx/bg.png);
	background-repeat: repeat-x;
}
#wrapper h1 {
	height: 90px;
	line-height: 90px;
	background-image: url(gfx/logo.png);
	background-repeat: no-repeat;
	background-position: top right;
	padding: 0px 25px 0px 25px;
	font-weight: normal;
	font-size: 24pt;
	letter-spacing: -2px;
	word-spacing: 5px;
	color: #336699;
}
#content {
	padding: 0px 25px 10px 25px;
}
p, table, pre, h2, h3, ul, ol, dl {
	margin: 0px 0px 15px 0px;
}
p.important {
	background-color: #EFDFBF;
	padding: 10px;
	font-size: 8pt;
}
p#submit, p#submit input {
	text-align: center;
	font-weight: bold;
}
p#submit input {
	padding: 5px;
}
h2 {
	color: #336699;
	font-weight: normal;
	font-size: 14pt;
	border-bottom: 1px solid silver;
}
h3 {
	color: #333;
	font-weight: bold;
	font-size: 10pt;
}
ul, ol {
```

```
        margin-left: 35px;
}
dl dt {
        font-weight: bold;
        color: #333;
}
dl dd {
        margin-left: 35px;
        margin-bottom: 5px;
}
table {
        background-color:#000000;
        border-collapse: collapse;
        margin-left: 0;
        margin-right: 0;
}
table th, table td {
        padding: 5px;
}
td {
        background-color:#cccc99;
}
table th {
        text-align: left;
        color: #336699;
}
table td.title {
        width: 135px;
}";
        echo "</style>\n";
if(!$_POST[user])                              //如果没有默认参数，则显示 HTML
{
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.user.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入注册用户名！ \");\n";
        echo "\t\ttheForm.user.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pass.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入用户密码！ \");\n";
        echo "\t\ttheForm.pass.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pass.value.length < 8 )\n";
        echo "\t{\n";
```

```
echo "\t\talert(\"密码至少要 8 位！ \");\n";
echo "\t\ttheForm.pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.re_pass.value !=theForm.pass.value)\n";
echo "\t{\n";
echo "\t\talert(\"确认密码与密码不一致！ \");\n";
echo "\t\ttheForm.re_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.nickname.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入昵称！ \");\n";
echo "\t\ttheForm.nickname.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "function s_photo(the)\n";
echo "{\n";
echo "\tdocument.img.src='images/'+the.photo.value+'.bmp';\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
echo "<tr>\n";
echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">注册新用户</font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户名：（后台登录）</td>\n";
echo "<td><input type=\"text\" name=\"user\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户密码：（不小于 8 位）</td>\n";
echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>确认密码：</td>\n";
echo "<td><input type=\"password\" name=\"re_pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户 E-mail：（可选）</td>\n";
echo "<td><input type=\"text\" name=\"email\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户昵称：（前台显示）</td>\n";
echo "<td><input type=\"text\" name=\"nickname\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
```

```
        echo "<td>选择性别：</td>\n";
        echo "<td>\n";
        echo "<input type=\"radio\" name=\"sex\" value=\"boy\" checked>男\n";
        echo "<input type=\"radio\" name=\"sex\" value=\"girl\">女\n";
        echo "</td>\n";
        echo "</tr>\n"
        echo "<tr>\n";
        echo "<td>选择图像：</td>\n";
        echo "<td>";
        echo "<select name=\"photo\" size=\"1\" onchange=\"s_photo(this.form)\">\n";
        for($i=1;$i<21;$i++)
        {
            echo "<option value=".$i.">".$i."</option>\n";;
        }
        echo "</select>";
        echo "<img src=\"images/1.bmp\" name=\"img\">";
        echo"</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>\n";
        echo "<input type=\"submit\" value=\"下一步\">\n";
        echo "<input type=\"reset\" value=\"重新填\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
}
```

以上为用户注册的前台显示页面。20.3.2 小节介绍用户注册的后台处理。

20.3.2　注册的后台处理

注册的后台处理先要获取用户输入的各项内容。其中登录用户名是最为重要的一项。因为该项要求具有唯一性，即同一个论坛不能有同名用户的出现，所以要对用户输入的用户名进行审核。这就会出现两种情况：（1）如果已经存在同名用户，则给出错误提示，并要求用户重新输入新的用户名；（2）如果没有同名用户，则把用户输入信息作为一条新的记录添加到用户表中。

下面给出具体的处理过程。请参看如下代码：

```
else                                            //如果有默认参数，则执行操作
{
        $user=$_POST[user];                     //获取用户输入数据
        $pass=md5($_POST[pass]);                //获取密码并进行 MD5 处理
        $email=$_POST[email];
        $nickname=$_POST[nickname];
```

```
        $sex=$_POST[sex];
        $time=date("Y 年 m 月 d 日");                              //获取当前时间
        $photo=$_POST[photo].".bmp";
        require "20-1.php";                                        //调用配置文件
        $sql="select id from $table_user where username='$user'";
        $result=mysql_query($sql,$link);                           //发送查找用户名的 SQL 请求
        $nums=mysql_num_rows($result);                             //获取查找结果
        if($nums!=0)                                               //如果结果不等于 0
        {
            echo "注册的用户名$user 已经存在！<p>";                 //给出相应提示
            echo "请点<a href=# onclick=history.go(-1)>这里</a>返回，重新输入新的用户名！";
            exit();                                                //退出所有 PHP 操作
        }
        else                                                       //如果结果为 0
        {
            $sql="insert                                                                into
    $table_members(name,password,email,nickname,sex,photo,reg_date)values('$user','$pass','$email','$nickname','$sex','$photo','$time')";
            mysql_query($sql,$link);                               //发送插入记录的 SQL 请求
            echo "新用户$user 注册成功！<p>";                       //显示成功提示
            echo "点<a href=20-4.php>这里</a>进行登录！";
        }
    }
    ?>
```

把以上代码与 20.3.1 小节的代码结合起来保存为“20-3.php”。然后在 PHP 执行环境下运行该 PHP 文件，首次运行会出现如图 20.3 所示的执行结果。

按照要求输入相应的用户、密码、确认密码、电子信箱、昵称及选择相应的图像后单击下一步按钮。如果用户名已经存在，则会出现如图 20.4 所示的提示。





图 20.3　注册新用户执行结果

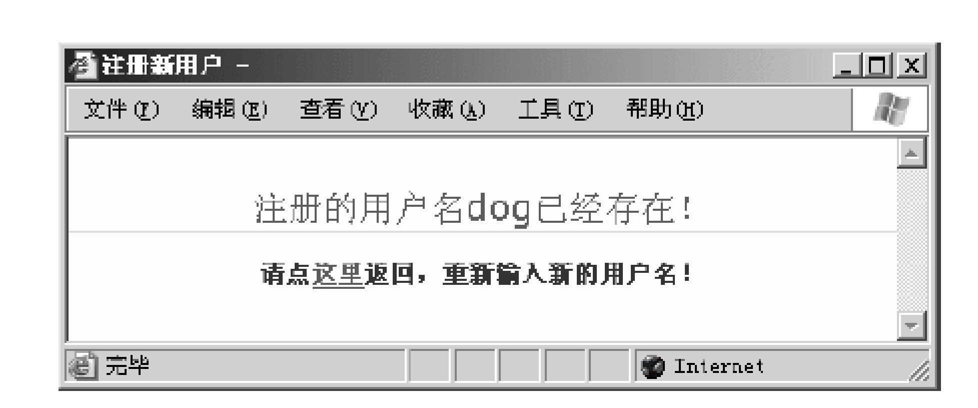


图 20.4　注册用户时用户名重复执行结果

如果输入的用户名是没有注册过的，则会出现如图 20.5 所示的执行结果。

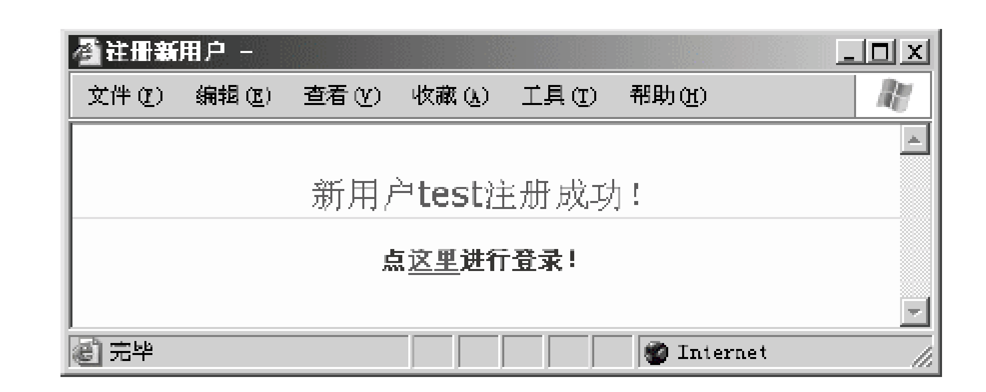


图 20.5　注册用户时注册成功执行结果

这时由于并不存在重名用户，所以允许用户注册，并将用户的输入信息写入用户列表中，并可以使用该用户登录系统了。

用户的注册及注册处理就介绍到这里。20.3.3 小节来讲解用户的登录与登录处理。

20.3.3　用户登录页面

用户信息添加进用户表后，就要让用户能够正确登录到系统中了。相比用户注册，用户登录就要简单一些。只需要给出用户名及密码就可以了。如果用户输入信息与库中存在的一致，则把用户名及相关信息存为 Cookie 变量。谈到 Cookie 在用户登录时也可以为用户提供一个现在比较流行的 Cookie 保存时间选择。这样就可以让用户自由选择 Cookie 的保留时间。如果是在公用计算机或者网吧使用论坛系统，就设为最短时效即关闭浏览器清除 Cookie。如果是在私有计算机或者家里使用，则可以设为一个月一年等更长的时间。

下面给出用户登录前台的具体代码：

```
<?php
if(!$_POST[user])                                   //如果没有默认参数，则显示 HTML
{
echo "<html>\n";
echo "<head>\n";
echo "<title>注册用户登录</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<style>\n";
echo "* {
padding: 0;
margin: 0;
}
body {
	font-family: verdana, sans-serif;
	font-size: 10pt;
	background-color: #FFFFEE;
	padding: 25px 0px 25px 0px;
}
a:link, a:active, a:visited {
	color: #336699;
	text-decoration: underline;
}
```

```
a:hover {
        color: #7F0000 !important;
        text-decoration: none;
}
select option {
        padding-right: 3px;
}
p, table, pre, h2, h3, ul, ol, dl {
        margin: 0px 0px 15px 0px;
}
h2 {
        color: #336699;
        font-weight: normal;
        font-size: 14pt;
        border-bottom: 1px solid silver;
}
h3 {
        color: #333;
        font-weight: bold;
        font-size: 10pt;
}
ul, ol {
        margin-left: 35px;
}
table {
        background-color:#000000;
        border-collapse: collapse;
        margin-left: 0;
        margin-right: 0;
}
table th, table td {
        padding: 5px;
}
td {
        background-color:#cccc99;
}
table th {
        text-align: left;
        color: #336699;
}
table td.title {
        width: 135px;
}";
        echo "</style>\n";
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.user.value == \"\")\n";
        echo "\t{\n";
```

```
        echo "\t\talert(\"请输入用户名！ \");\n";
        echo "\t\ttheForm.user.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pass.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入用户密码！ \");\n";
        echo "\t\ttheForm.pass.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "</script>\n";
        echo "<center>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">注册用户登录</font></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>用户名：</td>\n";
        echo "<td><input type=\"text\" name=\"user\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>用户密码：</td>\n";
        echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择 COOKIE 有效期：</td>\n";
        echo "<td>";
        echo "<select name=\"cook_t\" size=\"1\">\n";
        echo "<option value=\"1\">最短时效</option>";
        echo "<option value=\"2\">1 天</option>";
        echo "<option value=\"3\">1 月</option>";
        echo "<option value=\"4\">1 年</option>";
        echo "</select>";
        echo"</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>\n";
        echo "<input type=\"submit\" value=\"下一步\">\n";
        echo "<input type=\"reset\" value=\"重新填\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
}
```

上面给出的代码是用来显示注册用户登录的前台。20.3.4 小节来为读者介绍用户登录的后台处理。

20.3.4　登录出错及处理

用户登录的后台处理要先获取用户输入的数据，最主要是用户名与密码两项。然后对库中的用户数据表进行遍历看有没有相应的用户存在。如果存在相应的用户则把用户名写入 Cookie 变量，并根据用户的选择为 Cookie 设置生命期限，显示登录成功的提示，或者直接跳转到论坛的首页。反之，如果不存在相应的用户或者密码不正确则给出错误提示，给出返回上页的链接，让用户再次输入选择。

具体处理过程请参看如下代码：

```
else
{
	$user=$_POST[user];                                  //获取用户输入参数
	$pass=md5($_POST[pass]);
	$cook_t=$_POST[cook_t];
	require "20-1.php";
	$sql="select id from $table_members where name='$user' and password='$pass'";
	$result=mysql_query($sql,$link) or die(mysql_error());   //查找用户
	$nums=mysql_num_rows($result);                       //把查找记录数赋值给变量
	if($nums==0)                                         //如果记录数为 0，则显示内容
	{
		echo "<html>\n";
		echo "<head>\n";
		echo "<title>注册用户登录</title>\n";
		echo "</head>\n";
		echo "<body>\n";
		echo "<center>";
		echo "<h2>输入的用户名或者密码错误！</h2>";
		echo "<h3>请点<a href=# onclick=history.go(-1)>这里</a>返回重新输入！</h3>";
		echo "</center>";
		exit();

	}
	else                                                 //如果存在记录的处理
	{
		if($cook_t==1)                                   //根据用户选择注册 COOKIE
		{
			setcookie("user","$user");                   //最短
		}
		elseif($cook_t==2)
		{
			setcookie("user","$user",time()+60*60*24);   //一天
		}
		elseif($cook_t==3)
		{
			setcookie("user","$user",time()+60*60*24*30); //一月
		}
		else
```

```
            {
                setcookie("user","$user",time()+60*60*24*30*360);   //一年
            }
            echo "<html>\n";
            echo "<head>\n";
            echo "<title>注册用户登录</title>\n";
            echo "</head>\n";
            echo "<body>\n";
            echo "<center>";
            echo "<h2>用户".$user."登录成功！</h2>";
            echo "<h3>两秒后进入论坛主题页面！</h3>";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=20-5.php\">";
            echo "</center>";
        }
    }
    ?>
```

将以上代码与 20.3.3 小节所列的用户登录页面代码结合起来，保存为“20-4.php”。在 PHP 执行环境下运行该文件，执行结果如图 20.6 所示。按照要求输入相应的用户名、密码，单击“下一步”按钮开始进行登录。如果输入的用户或者密码错误，则会出现如图 20.7 所示的出错提示。



图 20.6　用户登录页面执行结果

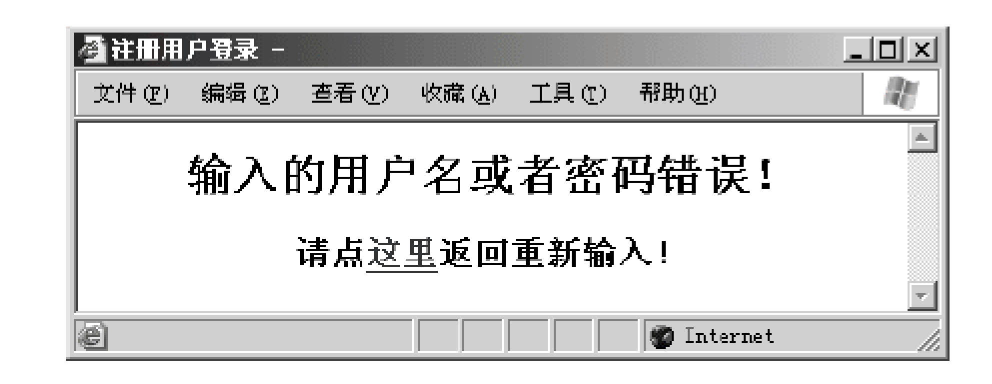


图 20.7　用户登录出错提示

如果输入的用户名及密码是库中已经存在的并且正确的话。登录程序将会自动跳转到论坛的显示首页 20-5.php。由于还没有编写这个页面，所以应该会出现无法找到所请求页面的提示。20.4 节就来着手编写该显示首页文件。

20.4　论坛首页、主论坛、分论坛显示文件的创建

论坛的表创建完毕，用户注册登录系统也都完善了。接下来，就开始论坛首页、主论坛、分论坛显示页面文件的创建。

20.4.1　论坛首页显示文件的创建

论坛首页文件是整个论坛系统中最为重要的文件。通常的论坛首页文件有以下作用：显示用户是

否登录，如果用户已经登录则显示登录用户名，反之则给出登录链接；显示所有主分类及其所属的子分类，某一分类的总贴子数，其中的主题数，最后发贴时间，最后的作者、最后一条贴子的标题等内容；通常还会在下方给出当前在线的用户（由于，本章所介绍的 BBS 系统没有设计在线用户，所以就没有这个功能）。

基于这些需要，一步步来创建该论坛首页文件。具体内容请参考以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>论坛首页</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<style>\n";
echo "body {
	font-family: verdana, sans-serif;
	font-size: 10pt;
	background-color: #FFFFEE;
	padding: 25px 0px 25px 0px;
}
a:link, a:active, a:visited {
	color: #0033ff;
	text-decoration: underline;
}
a:hover {
	color: #7F0000 !important;
	text-decoration: none;
}
select option {
	padding-right: 3px;
}
p, table, pre, h2, h3, ul, ol, dl {
	margin: 0px 0px 15px 0px;
}
p.important {
	background-color: #EFDFBF;
	padding: 10px;
	font-size: 8pt;
}
p#submit, p#submit input {
	text-align: center;
	font-weight: bold;
}
p#submit input {
	padding: 5px;
}
h2 {
	color: #336699;
	font-weight: normal;
	font-size: 14pt;
```

```
	border-bottom: 1px solid silver;
}
h3 {
	color: #333;
	font-weight: bold;
	font-size: 10pt;
}
ul, ol {
	margin-left: 35px;
}
dl dt {
	font-weight: bold;
	color: #333;
}
dl dd {
	margin-left: 35px;
	margin-bottom: 5px;
}
table {
	background-color:#000000;
	border-collapse: collapse;
	margin-left: 0;
	margin-right: 0;
}
table th, table td {
	padding: 5px;
}
td {
	background-color:#ddffff;
}
table th {
	text-align: left;
	color: #336699;
}
table td.title {
	width: 135px;
}";
echo "</style>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td>";
echo "<center><h2>论坛首页</h2></center>";
echo "</td>\n";
echo "</tr>\n";
echo "</table>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td>";
if(!$_COOKIE[user])                    //如果没有用户登录，则显示登录连接
```

```
{
	echo "<a href=\"20-4.php\">用户登录</a>";
}
else								//如果用户已经登录，则显示登录用户名
{
	require "20-1.php";
	$sql="select id from $table_members where name='$_COOKIE[user]'";
	$result=mysql_query($sql,$link);
	$rows=mysql_fetch_array($result);
	echo "登录用户：<a href=20-14.php?id=".$rows[0].">".$_COOKIE[user]."</a>";
}
echo "</td>\n";
echo "</tr>\n";
echo "</table>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td>论坛名称</td><td>论坛介绍</td><td>贴子数量</td><td>主题数量</td><td>最后贴子标题</td><td>最
后贴子回复时间</td>\n";
require "20-1.php";
$sql="select * from $table_topic where p_id=0";			//遍历贴子种类表显示主论坛
$result=mysql_query($sql,$link) or die(mysql_error());
while($rows=mysql_fetch_array($result))
{
	echo "<tr>\n";						//显示主论坛的标题及其介绍
	echo "<td colspan=\"6\"><a href=\"20-6.php?id=".$rows[id]."\">".$rows[topic_name]."</a> ：
".$rows[topic_description]."</td>\n";
	echo "</tr>\n";
	$temp2=$rows[id];
	$sql2="select * from $table_topic where p_id='$temp2'";	//遍历种类表显示主论坛下的分论坛
	$result2=mysql_query($sql2,$link) or die(mysql_error());
	while($rows2=mysql_fetch_array($result2))		//显示分论坛的各项信息
	{
		echo "<tr>\n";
		echo "<td><a href=\"20-7.php?id=".$rows2[id]."\">".$rows2[topic_name]."</a></td>\n";
		echo "<td>".$rows2[topic_description]."</td>\n";
		echo "<td>".$rows2[post_count]."</td>\n";
		echo "<td>".$rows2[post_m_count]."</td>\n";
		$sql3="select id,title,post_re_time from $table_posts where id='$rows2[last_post_id]'";
		$result3=mysql_query($sql3,$link) or die(mysql_error());
		$rows3=mysql_fetch_array($result3);		//显示最后一条贴子的相关信息
		echo "<td><a href=\"20-8.php?id=".$rows3[id]."\">".$rows3[title]."</a></td>\n";
		echo "<td>".$rows3[post_re_time]."</td>\n";
		echo "</tr>\n";
	}
	echo "<tr><td colspan=\"6\"> </td></tr>\n";
}
echo "</table>\n";
echo "</center>\n";
echo "</body>";
echo "</html>";
?>
```

将以上代码保存为“20-5.php”。由于系统在创建时已经创建了默认的主论坛和分论坛（详见 20.2.2 小节），所以目前显示时显示的只有 1 个默认的主论坛与其下的一个分论坛。执行结果如图 20.8 所示。

图 20.8　论坛首页文件首次执行结果

20.4.2　主论坛显示文件的创建

主论坛是论坛分类数据中一种比较特殊的类型。它代表着论坛的一级分类，在它的下面还有更详细的二级分类。做这样的划分是为了更详细的划分论坛结构，如一个普通的编程论坛可能包括技术区、娱乐区等内容。其中的技术区、娱乐区就相当于主论坛，而技术区里可能会划分为 PHP 编程区、JavaScript 编程区、MySQL 数据库编程区等二级分类；而娱乐区也可能会分为灌水区、贴图区、音乐电影区等二级分类。

既然论坛有这样的多级分类划分。所以显示主论坛及其下的分论坛是很有必要的。下面介绍如何用 PHP 代码来显示主论坛。

具体内容参考以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>主论坛显示</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])                                          //如果没有参数，则显示内容
{
    echo "<h2>没有请求 ID</h2>";
    echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
    exit();
}
else                                                    //如果存在参数，则显示相应论坛
{
    require "20-1.php";
    $sql="select p_id from $table_topic where id='$_GET[id]'";
    $result=mysql_query($sql,$link) or die(mysql_error());
    $rows=mysql_fetch_array($result);
```

```
        if($rows[0]!=0)                                        //如果论坛不是主论坛
        {
            echo "<h2>请求的 ID 不是主论坛</h2>";                 //显示信息
            echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
            exit();                                            //中止所有 PHP 执行
        }
        else                                                   //如果是主论坛，则显示 HTML
        {
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<tr>\n";
            echo "<td>";
            echo "<center><h2>主分类论坛首页</h2></center>";
            echo "</td>\n";
            echo "</tr>\n";
            echo "</table>\n";
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<tr>\n";
            echo "<td><a href=20-5.php>论坛首页</a></td>\n";
            echo "</tr>\n";
            echo "</table>\n";
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<tr>";
            echo "<td>论坛</td><td>贴数</td><td>主题</td><td>最后贴标题</td><td>最后贴回复时间</td>\n";
            echo "</tr>";
            $sql="select * from $table_topic where p_id='$_GET[id]'";
            $result=mysql_query($sql,$link) or die(mysql_error());
            while($rows=mysql_fetch_array($result))            //循环显示分论坛信息
            {
                echo "<tr>\n";
                echo "<td><a href=\"20-7.php?id=".$rows[id]."\">".$rows[topic_name]."</a><br>\n";
                echo $rows[topic_description]."</td>\n";
                echo "<td>".$rows[post_count]."</td>\n";
                echo "<td>".$rows[post_m_count]."</td>\n";
                $sql2="select id,title,post_re_time from $table_posts where id='$rows[last_post_id]'";
                $result2=mysql_query($sql2,$link) or die(mysql_error());
                $rows2=mysql_fetch_array($result2);            //显示最后一条贴子的相关信息
                echo "<td><a href=\"20-8.php?id=".$rows2[id]."\">".$rows2[title]."</a></td>\n";
                echo "<td>".$rows2[post_re_time]."</td>\n";
                echo "</tr>\n";
            }
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "</html>\n";
        }
    }
    ?>
```

把以上代码保存为“20-6.php”。单击图 20.8 中的“默认主类别 1”链接，将打开 20-6.php 并且带有参数 ID=1。其执行结果如图 20.9 所示。

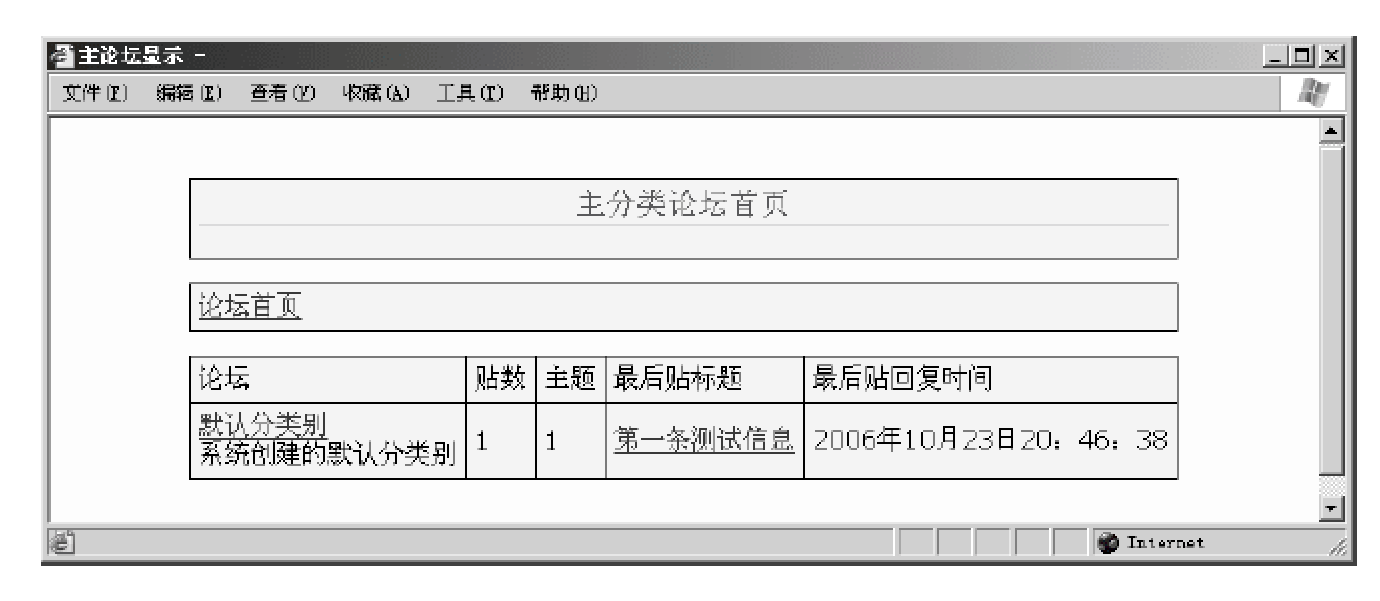


图 20.9　主论坛显示执行结果

与 20-5.php 不同的是，论坛首页显示文件将显示所有主分类论坛及其下面的分论坛的信息。而主论坛显示页则只显示本类分论坛的信息。

20.4.3　分论坛显示文件的创建

这一小节来创建分论坛的显示页面文件。该文件实现的作用有，显示当前分类论坛下的贴子信息，其包括标题、作者、查看、回复、最后发表等内容。下面就通过具体的代码来实现分论坛页面的显示。

具体内容请参看以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>分论坛显示</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])                                          //如果没有参数，则显示内容
{
    echo "<h2>没有请求 ID</h2>";
    echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
    exit();
}
else                                                    //如果有参数提示
{
    require "20-1.php";
    $sql="select p_id from $table_topic where id='$_GET[id]'";
    $result=mysql_query($sql,$link) or die(mysql_error());
    $rows=mysql_fetch_array($result);
    if($rows[0]==0)                                     //如果论坛不是分论坛
    {
        echo "<h2>请求的 ID 不是分论坛</h2>";           //显示内容
        echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
        exit();
    }
    else                                                //请求 ID 是分论坛显示 HTML
    {
```

```
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td>";
echo "<center><h2>分论坛首页</h2></center>";
echo "</td>\n";
echo "</tr>\n";
echo "</table>\n";
$sql="select p_id from $table_topic where id='$_GET[id]'";
$result=mysql_query($sql,$link) or die(mysql_error());
$rows=mysql_fetch_array($result);                    //获取主论坛 ID
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td><a href=20-5.php>论坛首页</a>|<a href=20-6.php?id=".$rows[0].">主论坛</a>";
echo "</td>\n";                                      //显示论坛首页及主论坛连接
if($_COOKIE[user])                                   //判断用户是否登录
{
    echo "<td align=right><a href=20-8.php?topic_id=".$_GET[id].">发表新主题</a>";
    echo "</td>\n";                                  //如果用户已经登录显示发贴连接
}
echo "</tr>\n";
echo "</table>\n";
$sql="select id from $table_posts where topic_id='$_GET[id]' and re_id=0";
$result=mysql_query($sql,$link);                     //获取主题数
$nums=mysql_num_rows($result);                       //获取主题总数
$p_count=ceil($nums/10);                             //求总页数
if ($_GET["page"]==0 && !$_GET["page"])              //如果没有请求页面
$page=1;                                             //当前页为第一页
else
$page=$_GET[page];                                   //获取当前页
$s=($page-1)*10+1;
$s=$s-1;                                             //当前页最多显示数
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>";
echo "<td>标题</td><td>作者</td><td>查看</td><td>回复</td><td>最后发表</td>\n";
echo "</tr>";
$sql="select * from $table_posts where topic_id='$_GET[id]' and re_id=0  order by post_re_time
desc limit $s, 10";
$result=mysql_query($sql,$link) or die(mysql_error());
while($rows=mysql_fetch_array($result))              //循环显示贴子信息
{
    echo "<tr>\n";
    echo "<td><a href=\"20-8.php?id=".$rows[id]."\">".$rows[title]."</a></td>\n";
    echo "<td>".$rows[poster]."</td>\n";
    echo "<td>".$rows[view_count]."</td>\n";
    echo "<td>".$rows[re_count]."</td>\n";
    echo "<td>".$rows[post_re_time]."</td>\n";
    echo "</tr>\n";
}
echo "</table>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
```

```
            echo "<tr>\n";
            echo "<td>";
            echo "<center>";
            $prev_page=$page-1;                                  //分页显示前一页
            $next_page=$page+1;                                  //定义分页显示下一页
            if ($page<=1)                                        //如果当前页小于等于 1
            {
                echo "第一页 | ";
            }
            else                                                 //如果当前页大于 1
            {
                echo "<a href='$_SERVER[PHP_SELF]?page=1'>第一页</a> | ";
            }
            if ($prev_page<1)                                    //如果前一页小于 1
            {
                echo "上一页 | ";
            }
            else                                                 //如果前一页大于等于 1
            {
                echo "<a href='$_SERVER[PHP_SELF]?page=$prev_page'>上一页</a> | ";
            }
            if ($next_page>$p_count)                             //如果下一页大于总页数
            {
                echo "下一页 | ";
            }
            else                                                 //如果下一页小于等于总页数
            {
                echo "<a href='$_SERVER[PHP_SELF]?page=$next_page'>下一页</a> | ";
            }
            if ($page>=$p_count)                                 //如果当前页大于等于总页数
            {
                echo "最后一页</p>\n";
            }
            else                                                 //如果当前页小于总页数
            {
                echo "<a href='$_SERVER[PHP_SELF]?page=$p_count'>最后一页</a></p>\n";
            }
            echo "</center>";
            echo "</td>\n";
            echo "</tr>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "</html>\n";
        }
    }
    ?>
```

将上面的代码保存为“20-7.php”。然后单击图 20.9 左下方的默认分类别就可以打开分论坛显示文件并且带有 ID=2，相当于执行 20-7.php?id=2。其执行结果如图 20.10 所示。

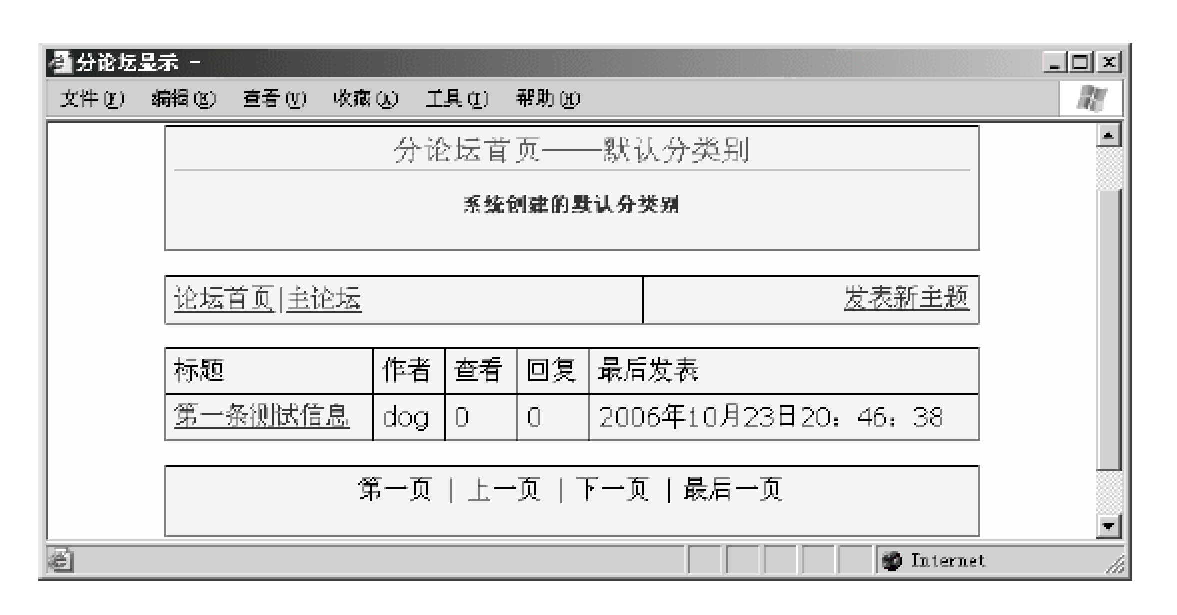


图 20.10　分论坛显示页面执行结果

从图 20.10 可以发现，显示出了本类论坛的所有主题，并且显示了主题的作者、查看次数、回复次数及最后回复时间。

论坛的显示部分就介绍到这里。20.5 节将开始介绍用户如何发表新主题及回复已经存在的主题。

20.5　主题的显示与回复

论坛的作用就是用户之间就某一主题进行讨论，所以主题的显示与回复是论坛所有实现功能中的重中之重。这一节介绍如何使用 PHP 代码实现发布新主题，以及现有主题的显示及回复功能。

20.5.1　发表新的主题

发表新的主题的实现方法是按照用户的输入将记录添加到贴子数据表。最为关键的一项是其 re_id 字段要为 0。这样就代表该条贴子是新的主题了。

下面给出具体代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>发表新主题</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_COOKIE[user])                                //用户没有登录的非法请求
{
    echo "<h2>匿名用户不允许发贴！ </h2>";
    echo "<h3>点<a href=20-4.php>这里</a>登录</h3>";
    exit();
}
if(!$_GET[topic_id])                               //没有分论坛参数
{
    echo "<h2>没有请求 ID</h2>";
    echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
    exit();
```

```
}
else                                                        //如果有请求分论坛 ID
{
    require "20-1.php";
    $sql="select p_id from $table_topic where id='$_GET[topic_id]'";
    $result=mysql_query($sql,$link) or die(mysql_error());
    $rows=mysql_fetch_array($result);
    if($rows[0]==0)                                         //如果请求 ID 不是分论坛
    {
        echo "<h2>请求的 ID 不是分论坛</h2>";
        echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
        exit();
    }
    else                                                    //如果是分论坛
    {
        if(!$_POST[title])                                  //如果没有提交变量，则显示前台
        {
            if(!$_GET[re_id]) $re_id=0;                     //如果没有回复 ID 号
            else $re_id=$_GET[re_id];                       //获取回复 ID 号
            echo "<script language=\"javascript\">\n";
            echo "function juge(theForm)\n";
            echo "{\n";
            echo "\tif (theForm.title.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入主题名称！ \");\n";
            echo "\t\ttheForm.title.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
            echo "\tif (theForm.content.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入主题内容！ \");\n";
            echo "\t\ttheForm.content.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
            echo "}\n";
            echo "</script>\n";
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<tr>\n";
            echo "<td>";
            echo "<center><h2>";
            if($re_id==0) echo "发表新主题";
            else echo "发表回复";
            echo "</h2></center>";
            echo "</td>\n";
            echo "</tr>\n";
            echo "</table>\n";
            $sql="select p_id,topic_name from $table_topic where id='$_GET[topic_id]'";
            $result=mysql_query($sql,$link) or die(mysql_error());
            $rows=mysql_fetch_array($result);
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
```

```
            echo "<tr>\n";
            echo "<td><a href=20-5.php>论坛首页</a>|<a href=20-6.php?id=".$rows[0].">主论坛</a>|<a
href=20-7.php?id=".$_GET[topic_id].">".$rows[1]."</a></td>\n";
            echo "</tr>\n";
            echo "</table>\n";
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<form method=\"post\" action=\"20-8.php?topic_id=".$_GET[topic_id]."\"
onsubmit=\"return juge(this)\">\n";
            echo "<input type=hidden name=re_id value=".$re_id.">";
            echo "<tr>\n";
            echo "<td>输入标题：</td>\n";
            echo "<td><input type=text name=title></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td>输入内容：</td>\n";
            echo "<td><textarea rows=5 cols=50 name=content></textarea></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td colspan=\"2\"><center><input type=submit value=提交><input type=reset value=重置
></center></td>\n";
            echo "</tr>\n";
            echo "<form>";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "</html>\n";
        }
        else                                            //后台处理
        {
            $title=$_POST[title];                       //获取参数
            $content=$_POST[content];
            $re_id=$_POST[re_id];
            $topic_id=$_GET[topic_id];
            $ip=$_SERVER[REMOTE_ADDR];
            $time=date("Y 年 m 月 d 日 G：i：s");
            $sql="select id,nickname from $table_members where name='$_COOKIE[user]'";
            $result=mysql_query($sql,$link);
            $rows=mysql_fetch_array($result);
            $sql="insert                                                              into
$table_posts(topic_id,re_id,poster_id,poster_ip,poster,title,content,post_time,post_re_time)values('$topic_id','$r
e_id','$rows[0]','$ip','$rows[nickname]','$title','$content','$time','$time')";
            if(mysql_query($sql,$link) or die(mysql_error()))
            {
                if($re_id==0)                           //如果是发表主题的操作
                {
                    $sql="select max(id) from $table_posts";
                    $result=mysql_query($sql,$link);
                    $rows=mysql_fetch_array($result);
                    echo $rows[0];
                    $sql="update $table_posts set re_count=re_count+1,post_re_time='$time' where
```

```
id='$topic_id'";
                    mysql_query($sql,$link);
                }
                else                                    //如果是发表回复的操作
                {
                    $sql="update $table_posts set re_count=re_count+1 where id='$topic_id'";
                    mysql_query($sql,$link);
                }
                $sql="update $table_topic set post_count=post_count+1 where id='$topic_id'";
                mysql_query($sql,$link);
                $sql="update $table_topic set post_count=post_count+1 where id='$tipic_id'";
                mysql_query($sql,$link);
                echo "添加记录成功，现在返回分论坛页";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-7.php?id=".$topic_id."\">";
            }
            else
            {
                echo "添加记录时出错，现在返回";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-7.php?id=".$topic_id."\">";
            }
        }
    }
}
?>
```

把以上代码保存为“20-8.php”。单击图 20.10 中“发表新主题”的链接就相当于执行 20-8.php，其执行结果如图 20.11 所示。按照要求填入标题及内容。单击“提交”按钮就可以发表新的主题了。当主题成功发布后，程序会自动跳转到相应的子分类页面中，如图 20.12 所示。



图 20.11　发表新主题执行结果

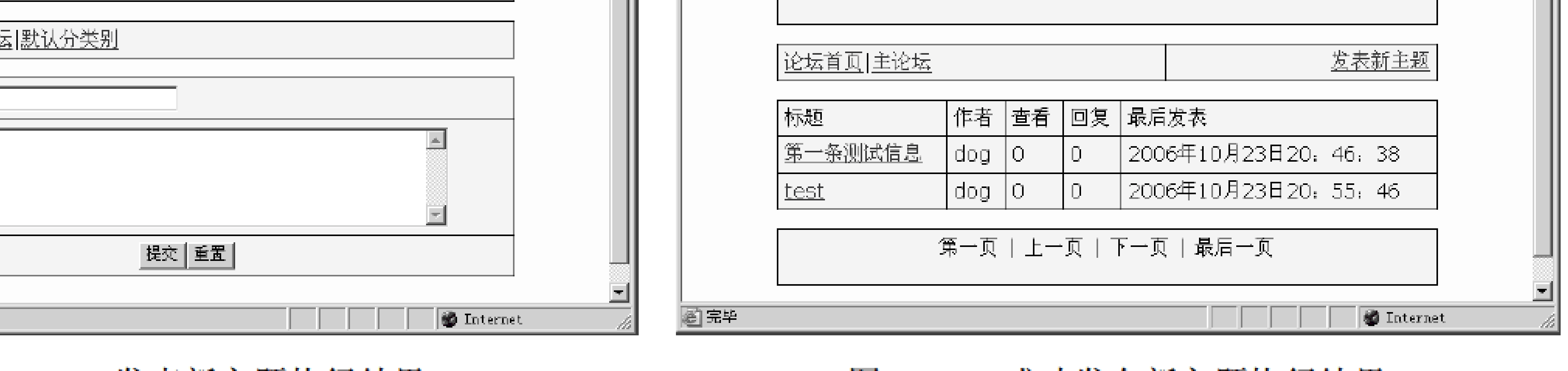


图 20.12　成功发布新主题执行结果

这样新的主题就成功添加到贴子的表中了。主题成功入库还是不够的，必须要让浏览者能够浏览才行。20.5.2 小节介绍如何显示已经存在主题。

20.5.2　现有主题的显示

这一小节来讲解如何显示已经存在的主题。主题的显示也很简单，首先获取用户提交的 ID，然后

判断该 ID 所对应的贴子是否是主题。因为贴子存在主题与主题的回复两种情况，所以要做这样的判断。如果提交的 ID 不是主题则给出相应提示，如果是主题，则显示该主题内容及对于该主题的回复。

这里涉及到对多个表的查询与读取。既要读取贴子相关内容，也要读取贴子对应的分类数据表记录的读取。另外，在浏览主题的同时要记得每请求一次主题显示文件，就要给主题的浏览次数即 view_count 增加 1。这样才能准确的获得主题的浏览次数。

下面给出主题显示的详细代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>显示主题</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])                                          //如果没有请求主题 ID
{
    echo "<h2>没有请求 ID</h2>";
    echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
    exit();
}
else                                                    //显示主题内容
{
    require "20-1.php";
    $sql="select * from $table_posts where id='$_GET[id]'";
    $result=mysql_query($sql,$link);
    $rows=mysql_fetch_array($result);
    if($rows[re_id]!=0)                                 //如果请求 ID 不是主题
    {
        echo "<h2>请求的贴子不是主题！ </h2>";
        echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
        exit();
    }
    else
    {
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<tr>\n";
        echo "<td>";
        echo "<center><h2>主题显示</h2></center>";
        echo "</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        $sql2="select id,p_id,topic_name from $table_topic where id='$rows[topic_id]'";
        $result2=mysql_query($sql2,$link);
        $rows2=mysql_fetch_array($result2);
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<tr>\n";
        echo  "<td><a  href=20-5.php> 论 坛 首 页 </a>|<a  href=20-6.php?id=".$rows2[1]."> 主 论 坛 </a>|<a
```

```
href=20-7.php?id=".$rows2[0].">".$rows2[2]."</a>";
        echo "</td>\n";
        echo   "<td   align=right><a   href=20-8.php?topic_id=".$rows2[0]."> 发 表 新 主 题 </a>|<a
href=20-8.php?topic_id=".$rows2[0]."&re_id=".$_GET[id].">发表回复</a>";
        echo "</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<tr>\n";
        echo "<td>\n";
        echo "作者：".$rows[poster];
        echo "<br>作者 ID：".$rows[poster_id];
        echo "<br>作者 IP：".$rows[poster_ip];
        echo "<br>标题：".$rows[title];
        echo "<br>内容：".$rows[content];
        echo "<br>发表时间：".$rows[post_time];
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>以下为该主题的回复</td>\n";
        echo "</tr>\n";
        $sql="select * from $table_posts where re_id='$_GET[id]'";
        $result=mysql_query($sql,$link);
        while($rows=mysql_fetch_array($result))
        {
            echo "<tr>\n";
            echo "<td>\n";
            echo "作者：".$rows[poster];
            echo "<br>作者 ID：".$rows[poster_id];
            echo "<br>作者 IP：".$rows[poster_ip];
            echo "<br>标题：".$rows[title];
            echo "<br>内容：".$rows[content];
            echo "<br>发表时间：".$rows[post_time];
            echo "</td>\n";
            echo "</tr>\n";
        }
        $sql="update $table_posts set view_count=view_count+1 where $id='$_GET[id]'";
        mysql_query($sql,$link);
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "</html>";
    }
}
?>
```

把以上代码保存为“20-9.php”。单击图 20.12 中贴子标题的链接就可以进入相应的主题显示页面了。单击“第一条测试信息”链接，如图 20.13 所示。

从图 20.13 可以发现，程序正确的显示出了相应主题及发贴者的相关信息。下面介绍如何对主题进

行回复。

图 20.13　主题显示执行结果

20.5.3　主题的回复

对主题进行回复，只需要给 20-8.php 加上相应的参数即可实现。在编写 20-8.php 时曾用到这样的参数 re_id。当该参数为 0 时，表示发表新的主题，而当该值为某数值时说明是对 ID 号为该值的主题进行回复。在这里直接调用即可实现对主题的回复。

不过回复的操作并不只是把用户输入记录添加进表就行了，还要进行一系列的操作，如使分类表中相应的记录的贴子数自增 1、使相应用户的发贴数自增 1 和相应主题的回复数自增 1 等。

单击图 20.13 中的“发表回复”链接，将打开 20-8.php。不过不同的是已经不是发表新主题了，而是回复主题了。填写相应的标题及内容，提交回复后，将回到分论坛页，单击回复的主题以查看回复是否正确显示，执行结果如图 20.14 所示。



图 20.14　回复过的主题显示页执行结果

显示回复时，只需要把回复内容跟在主题后面即可。从图 20.14 可以看出正确的显示了已经存在的主题内容及相应的回复。

关于主题贴子的显示与回复，这一节就介绍到这里。接下来的几节重点介绍管理员或者版主才有的权限的内容，如对整个论坛分类的管理、对贴子的管理等内容。

20.6　论坛分类的管理

系统在进行安装的时候创建了默认的主类别及分类别，但这对于一个通用的论坛来说是远远不够的。应该给予管理员相应的权限，以创建新的类别。这一节就来讲讲如何对论坛的分类进行管理，其中包括为论坛增加新的类别，及更改现有的类别。通常情况下，一个论坛分类在创建后就不能被删除了，所以就不再讨论如何删除分类的问题。

20.6.1　为论坛增加新的类别

为论坛增加类别包括两种情况：一种是增加主分类即主论坛，一种是为某一存在的主分类增加二级分类即分论坛。在有些比较大型的论坛程序中，甚至还有三级或者多级的分类。这里只探讨二级的情况。

为论坛增加新的类别的操作分几步进行。第 1 步先让用户选择欲创建的类别是主类别还是分类别。第 2 步，如果用户选择的是主类别，在这一步直接输入类别名称及介绍等相应信息即可；如果用户在第 1 步选择的是分类别，则还需要让用户选择，该分类属于已经存在的哪一个主类别。然后再按照选择输入类别名称、类别说明等内容。

下面给出具体的代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>增加新论坛</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
require "20-1.php";
$sql="select admin from $table_members where name='$_COOKIE[user]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows['admin']<3)
{
	echo "<h2>你没有这项权限来运行该文件！</h2>";
	echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
	exit();
}
else
```

```
{
	if(!$_POST[topic])
	{
		echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
		echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\">\n";
		echo "<tr>\n";
		echo "<td colspan=\"2\"><center><h2>创建论坛分类第一步</h2></center></td>\n";
		echo "</tr>";
		echo "<tr>\n";
		echo "<td>选择创建类别</td>\n";
		echo "<td>";
		echo "<select size=\"1\" name=\"topic\">\n";
		echo "<option value=\"1\">主类别</option>\n";
		echo "<option value=\"2\">分类别</option>\n";
		echo "</select>\n";
		echo "</td>\n";
		echo "</tr>";
		echo "<tr>\n";
		echo "<td colspan=\"2\"><center><input type=submit value=\"下一步\"></td>\n";
		echo "</tr>\n";
		echo "</form>\n";
		echo "</table>\n";
		echo "</center>\n";
		echo "</body>";
		echo "</html>\n";
	}
	else if(!$_POST[topic_name])
	{
		echo "<script language=\"javascript\">\n";
		echo "function juge(theForm)\n";
		echo "{\n";
		echo "\tif (theForm.topic_name.value == \"\")\n";
		echo "\t{\n";
		echo "\t\talert(\"请输入类别名称！ \");\n";
		echo "\t\ttheForm.topic_name.focus();\n";
		echo "\t\treturn (false);\n";
		echo "\t}\n";
		echo "\tif (theForm.topic_description.value == \"\")\n";
		echo "\t{\n";
		echo "\t\talert(\"请输入类别介绍！ \");\n";
		echo "\t\ttheForm.topic_description.focus();\n";
		echo "\t\treturn (false);\n";
		echo "\t}\n";
		echo "}\n";
		echo "</script>\n";
		echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
		echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\" onsubmit=\"return 
juge(this)\">\n";
		echo "<tr>\n";
		echo "<td colspan=\"2\"><center><h2>创建论坛分类第二步</h2></center></td>\n";
```

```
            echo "</tr>";
            echo "<input type=\"hidden\" name=\"topic\" value=\"".$_POST[topic]."\">";
            if($_POST[topic]==2)
            {
                echo "<tr>\n";
                echo "<td>选择分类别所属主类</td>\n";
                echo "<td>";
                echo "<select size=\"1\" name=\"p_id\">\n";
                $sql="select id,topic_name from $table_topic where p_id=0";
                $result=mysql_query($sql,$link);
                while($rows=mysql_fetch_array($result))
                {
                    echo "<option value=\"".$rows[id]."\">".$rows[topic_name]."</option>\n";
                }
                echo "</select>\n";
                echo "</td>\n";
                echo "</tr>";
            }
            echo "<tr>\n";
            echo "<td>输入类别名称</td>\n";
            echo "<td>";
            echo "<input type=\"text\" name=\"topic_name\">\n";
            echo "</td>\n";
            echo "</tr>";
            echo "<tr>\n";
            echo "<td>输入类别介绍</td>\n";
            echo "<td>";
            echo "<input type=\"text\" name=\"topic_description\">\n";
            echo "</td>\n";
            echo "</tr>";
            echo "<tr>\n";
            echo    "<td    colspan=\"2\"><center><input    type=button    value=\"   上   一   步   \"
    onclick=\"history.go(-1)\"><input type=submit value=\"下一步\"></td>\n";
            echo "</tr>\n";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>";
            echo "</html>\n";
        }
        else
        {
            $topic=$_POST[topic];
            $topic_name=$_POST[topic_name];
            $topic_description=$_POST[topic_description];
            if($topic==2)
            {
                $p_id=$_POST[p_id];
            }
            $sql="insert                                                                      into
    $table_topic(p_id,topic_name,topic_description)values('$p_id','$topic_name','$topic_description')";
```

```
            if(mysql_query($sql,$link))
            {
                echo "增加新论坛操作成功，现在返回首页！";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-5\">";
            }
            else
            {
                echo "增加新论坛操作失败，现在返回！";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-10\">";
            }
        }
    }
    ?>
```

把以上代码保存为“20-10.php”。直接在 PHP 执行环境下运行该文件。如果没有相应的权限，则会出现如图 20.15 所示的提示，而如果已经以管理员身份进行登录，重新执行该 PHP 文件，其执行结果会如图 20.16 所示。

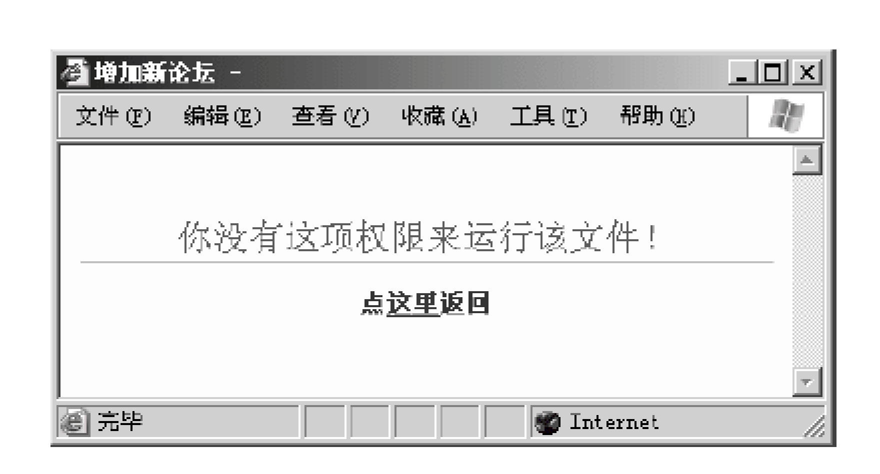


图 20.15　无权限时执行增加类别文件执行结果



图 20.16　以管理员身份执行添加操作第 1 步

这一步很简单，只需要选择相应的类别即可，这里选择创建分类别，然后单击“下一步”按钮则会出现如图 20.17 所示的画面。

由于要创建的是分类别，所以要选择分类别所属的主类别。但目前论坛分类数据表中只有一个主类别，所以不用选择，然后输入类别相应名称，及简单介绍即可。单击“下一步”按钮，如果成功执行的话，程序会在执行完相应的操作后直接跳转到论坛的首页。也可以通过首页的显示，查看是否添加成功。如图 20.18 所示是执行结果。

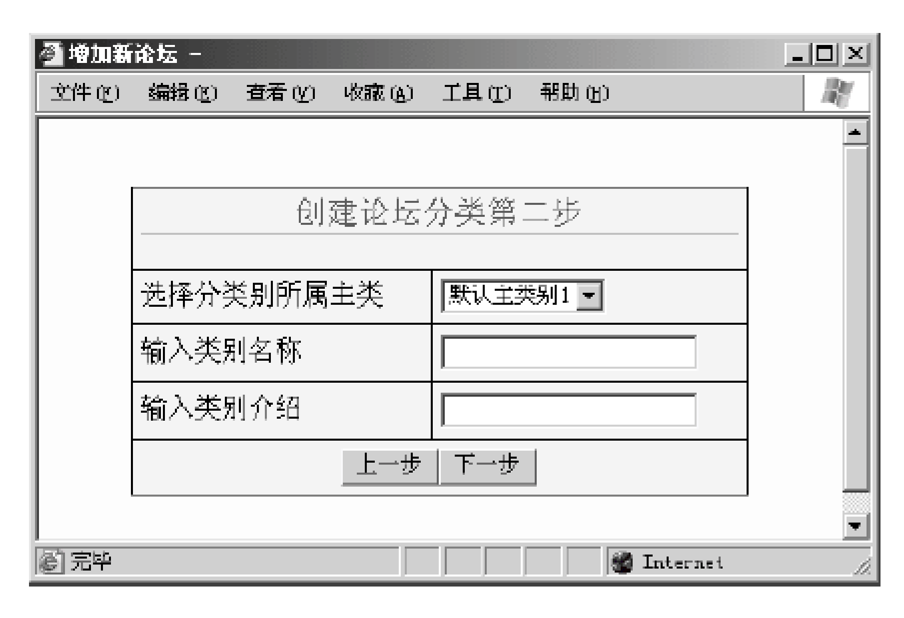


图 20.17　以管理员身份执行添加操作第 2 步

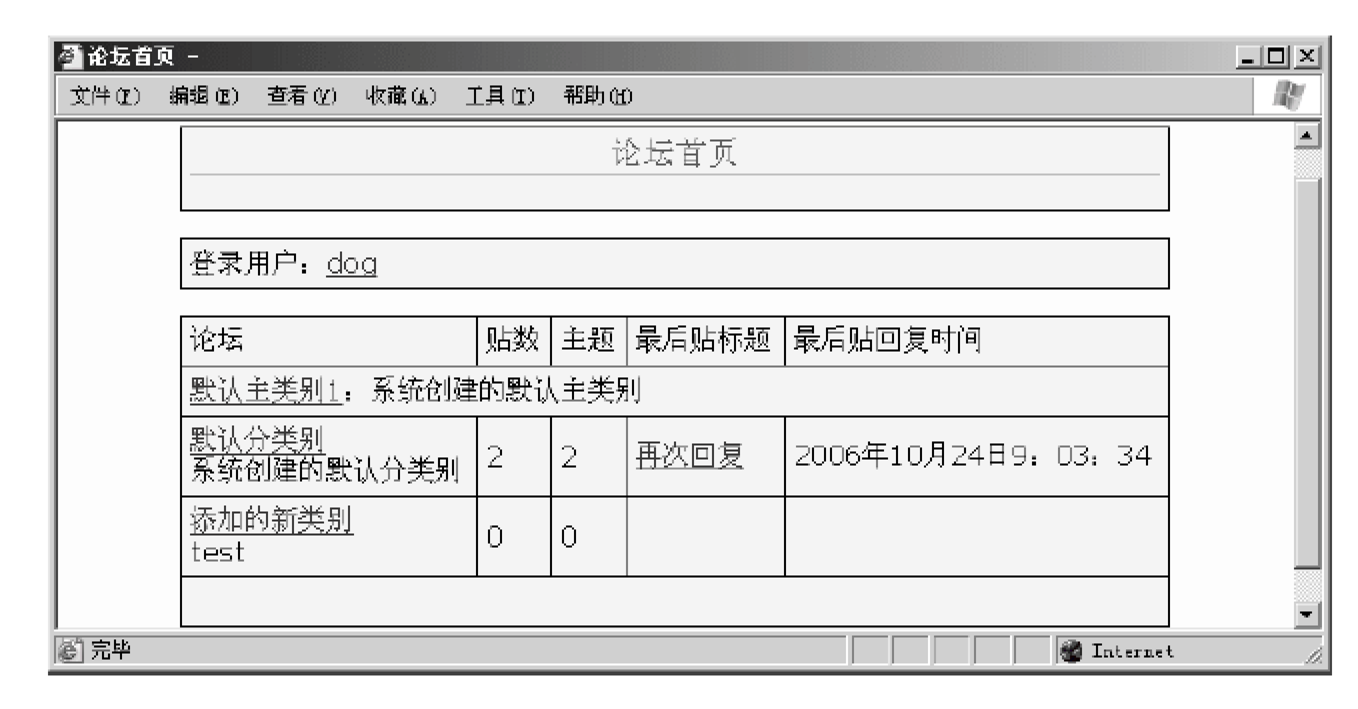


图 20.18　成功添加相应类别后的首页

比较图 20.18 与图 20.8 可以发现：第 1 个分类别的贴子除在数量上有区别之外还有一个区别就是多出了一个新的分类别。说明添加操作成功执行。

有一点要引起注意，由于用户发贴都是基于分类别发贴的，所以如果一个主类别下边没有相应的分类别，它是不会被显示的。所以，在添加完一个主类别后必须为其添加相应的分类别。

20.6.2　更改现有类别

论坛的分类在创建后不能一成不变。最起码在创建初期应该会有改变，如系统默认的类别并不符合用户使用的要求，所以就要改变相应类别的名称或者类别简介等内容。这一小节介绍对现有类别的修改操作。

由于某一类别的主题数及贴子数这些属性是动态生成的，并与论坛的贴子数据表是紧密相连的，所以，不能修改这些内容。只能修改类别名称、类别简介及分类别所属主类这些内容。

下面给出具体的操作代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改已有的论坛</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
require "20-1.php";
$sql="select admin from $table_members where name='$_COOKIE[user]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[admin]<3)
{
	echo "<h2>您没有这项权限来运行该文件！</h2>";
	echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
	exit();
}
else
{
	if(!$_POST[id])
	{
		echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
		echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\">\n";
		echo "<tr>\n";
		echo "<td colspan=\"4\"><center><h2>修改论坛第一步</h2></center></td>\n";
		echo "</tr>";
		echo "<tr>\n";
		echo "<td>选择论坛</td>\n";
		echo "<td>主/分</td>\n";
		echo "<td>论坛名称</td>\n";
		echo "<td>论坛简介</td>\n";
```

```
        echo "</tr>";
        $sql="select id,p_id,topic_name,topic_description from $table_topic";
        $result=mysql_query($sql,$link);
        while($rows=mysql_fetch_array($result))
        {
            echo "<tr>\n";
            echo "<td><input type=\"radio\" name=\"id\" value=\"".$rows[id]."\"></td>\n";
            echo "<td>";
            if($rows[p_id]==0)
            {
                echo "主论坛";
            }
            else
            {
                echo "分论坛";
            }
            echo "</td>\n";
            echo "<td>".$rows[topic_name]."</td>\n";
            echo "<td>".$rows[topic_description]."</td>\n";
            echo "</tr>\n";
        }
        echo "<tr>\n";
        echo "<td colspan=\"4\"><center><input type=submit value=\"下一步\"></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>";
        echo "</html>\n";

    }
    else if(!$_POST[topic_name])
    {
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.topic_name.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入类别名称！\");\n";
        echo "\t\ttheForm.topic_name.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.topic_description.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入类别介绍！\");\n";
        echo "\t\ttheForm.topic_description.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
```

```
        echo "</script>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\" onsubmit=\"return
juge(this)\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><h2>创建论坛分类第二步</h2></center></td>\n";
        echo "</tr>";
        echo "<input type=\"hidden\" name=\"id\" value=\"".$_POST[id]."\">";
        $sql="select * from $table_topic where id='$_POST[id]'";
        $result=mysql_query($sql,$link);
        $rows=mysql_fetch_array($result);
        if($rows[p_id]!=0)
        {
            echo "<tr>\n";
            echo "<td>选择分类别所属主类</td>\n";
            echo "<td>";
            echo "<select size=\"1\" name=\"p_id\">\n";
            $sql2="select id,topic_name from $table_topic where p_id=0";
            $result2=mysql_query($sql2,$link);
            while($rows2=mysql_fetch_array($result2))
            {
                echo "<option value=\"".$rows2[id];
                if($rows2[id]==$rows[p_id]) echo " checked ";
                echo "\">".$rows2[topic_name]."</option>\n";
            }
            echo "</select>\n";
            echo "</td>\n";
            echo "</tr>";
        }
        echo "<tr>\n";
        echo "<td>输入类别名称</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"topic_name\" value=\"".$rows[topic_name]."\">\n";
        echo "</td>\n";
        echo "</tr>";
        echo "<tr>\n";
        echo "<td>输入类别介绍</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"topic_description\" value=\"".$rows[topic_description]."\">\n";
        echo "</td>\n";
        echo "</tr>";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><input type=button value=\" 上 一 步 \"
onclick=\"history.go(-1)\"><input type=submit value=\"下一步\"></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>";
```

```
            echo "</html>\n";
        }
        else
        {
            $id=$_POST[id];
            $topic=$_POST[topic];
            $topic_name=$_POST[topic_name];
            $topic_description=$_POST[topic_description];
            if($_POST[p_id])
            {
                $p_id=$_POST[p_id];
            }
            else
            {
                $p_id=0;
            }
            $sql="update                                        $table_topic                    set
    p_id='$p_id',topic_name='$topic_name',topic_description='$topic_description' where id=$id";
            if(mysql_query($sql,$link))
            {
                echo "修改论坛操作成功，现在返回首页！";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-5\">";
            }
            else
            {
                echo "修改论坛操作失败，现在返回！";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-10\">";
            }
        }
    }
    ?>
```

将上述代码保存为“20-11.php”。与增加新类别操作一样，如果没有相应的权限，执行该文件将出现如图 20.15 一样的提示。如果以管理员身份运行该文件，其执行结果将如图 20.19 所示。

这里显示出了目前论坛中的所有一级论坛及二级论坛。选中前面的单选按钮以选择相应的记录进行操作。这里选择对默认分类别进行修改操作。选择好后，单击“下一步”按钮，将出现图 20.20 所示的执行结果。

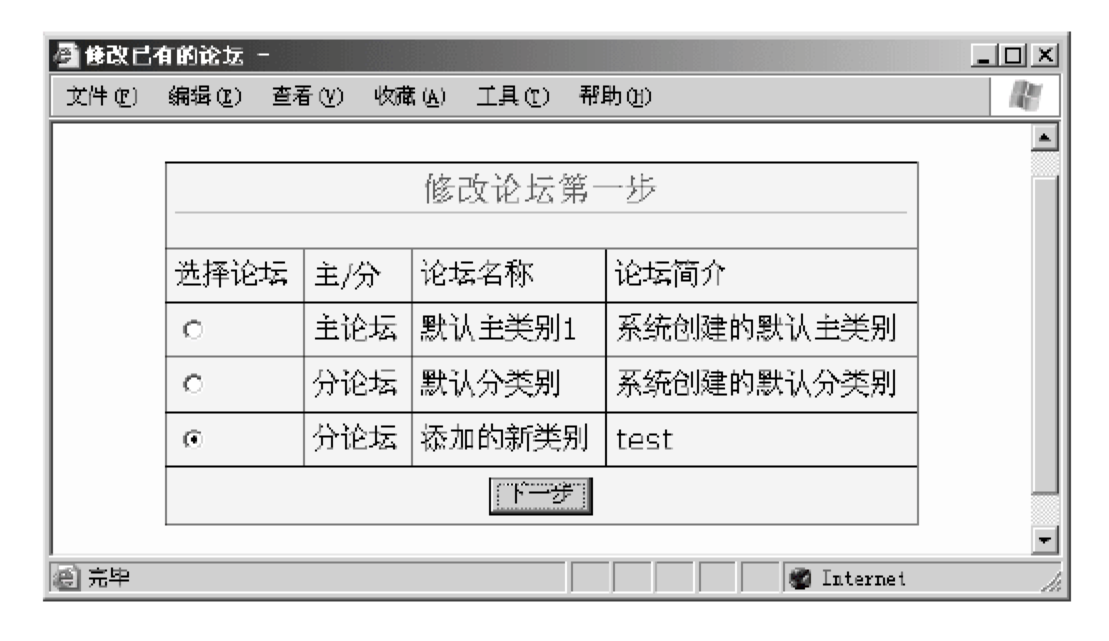


图 20.19　以管理员身份执行修改操作第一步



图 20.20　以管理员身份执行修改操作第二步

从图 20.20 可以看出，这里能够对分类别的所属主类别（如果操作的是主类别将没有这一项）、类别名称、类别介绍等内容进行修改。这里把类别名称更改为“PHP 编程”，类别介绍修改为“PHP 编程专区，欢迎各位 PHP 菜鸟大虾光临！”。由于现在只有一个主类别，所以主类别必须选择“默认主类别 1”选项。然后单击“下一步”按钮，如果操作成功，程序将自动跳转到论坛首页，如图 20.21 所示。

从图 20.21 可以看出相应的类别已经被修改了。同理，也可以对主类别或者其他所有类别进行相应修改。下面把“默认的主类别 1”修改为“网络编程区”，把类别介绍修改为“网络语言编程区：有 PHP、ASP、JavaScript 等内容”。修改后的结果如图 20.22 所示。



图 20.21　成功修改相应类别后的首页

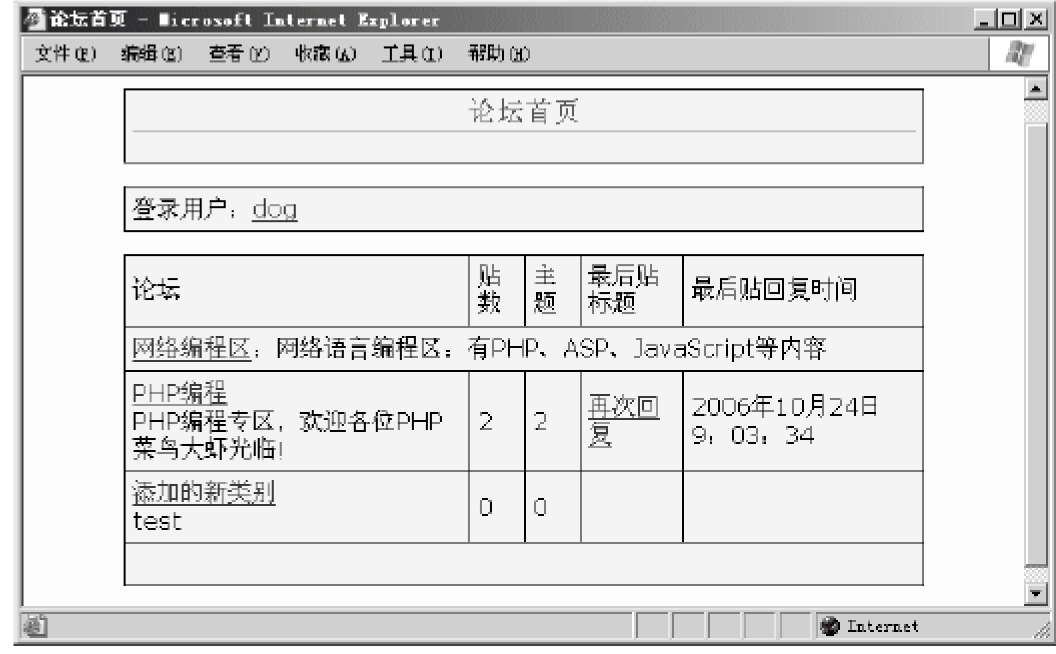


图 20.22　主类别修改之后的首页显示效果

通常情况下，不需要对论坛分类进行删除操作，所以这里要介绍的论坛分类就不再探讨分类的删除问题。至此，论坛分类管理就为读者介绍完毕。从这两节的介绍可以看出，对分类的管理，实际上还是对数据库表的添加或者修改操作，只要明白其中的道理，一切都变得很简单。

20.7　贴子的管理

这一节介绍对论坛中贴子的管理。贴子是构成论坛的灵魂，论坛只有有了好贴子才能发挥其生命力。那些广告贴、垃圾贴、含有不合法内容的贴子要及时清除。这一节的内容就是如何对贴子进行管理。其中包括贴子的编辑、非法贴的删除和如何防止用户恶意掘墓等内容。

20.7.1　编辑贴子

用户发的贴中，有时会有一些不合适的内容，或者用户发错的内容，这就应该给管理员和贴子的作者有编辑贴子内容的权限。这里要对显示贴子的页面（20-9.php）做一定改动。给相应的贴子上显示出编辑链接。当用户有编辑权限时（这里有两种情况，一种是用户本身是论坛管理员或者相应区的版主；另一种是这条贴子的作者就是浏览的用户），显示该编辑链接，反之不显示。这样，就可以实现贴子的编辑了。

具体怎么改动呢？

下面给出具体的代码。

分别找到显示主题与回复代码中的这样的内容：

```
echo "<td>";
echo "标题：".$rows[title];
echo "</td>";
```

把上面的内容修改如下：

```
echo "<td>";
echo "标题：".$rows[title];
if($rows[poster]==$_COOKIE[user])        //判断用户与贴子作者是否是同一人
echo "    <a href=\"20-12.php?id=".$rows[id]."\">编辑</a>";
echo "</td>";
```

这样当浏览的用户与发贴（或者回复）的作者是同一个人的话，就是在贴子或者回复的标题后会显示出“编辑”的链接，以方便用户对贴子实现编辑操作。

也可以这样：

```
echo "<td>";
echo "标题：".$rows[title];
$sql_t="select admin from $table_members where name='$_COOKIE[user]'";
$result_t=mysql_query($sql_t,$link);
$rows_t=mysql_fetch_array($result_t);
if($rows_t[admin]==3)
echo "    <a href=\"20-12.php?id=".$rows[id]."\">编辑</a>";
echo "</td>";
```

上面的修改是让管理员也有编辑贴子的权限，与普通用户不同。普通用户只能修改自己所发的贴子，而管理员则可以修改所有的贴子。

经过以上修改，再次运行 20-9.php 时的执行结果会如图 20.23 所示。



图 20.23　加入编辑链接的主题显示页面

经过以上的修改，对贴子进行修改的准备工作就算做完了。下面介绍如何修改贴子的内容。

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>编辑贴子</title>\n";
echo "</head>\n";
echo "<body>\n";
```

```
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])
{
	echo "<h2>没有请求 ID</h2>";
	echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
	exit();
}
else
{
	require "20-1.php";
	$sql="select poster,topic_id,re_id,title,content from $table_posts where id='$_GET[id]'";
	$result=mysql_query($sql,$link) or die(mysql_error());
	$rows=mysql_fetch_array($result);
	$sql2="select admin from $table_members where name='$_COOKIE[user]'";
	$result2=mysql_query($sql2,$link) or die(mysql_error());
	$rows2=mysql_fetch_array($result2);
	if(($rows[0]!=$_COOKIE[user]) or ($rows2[0]<3))		//如果用户不是发贴者或者不是管理员
	{
		echo "<h2>没有这样的权限</h2>";
		echo "<h3>点<a href=\"#\" onclick=\"history.go(-1)\">这里</a>返回</h3>";
		exit();
	}
	else
	{
		if(!$_POST[title])					//如果没有发送标题，则显示 HTML
		{
			echo "<script language=\"javascript\">\n";
			echo "function juge(theForm)\n";
			echo "{\n";
			echo "\tif (theForm.title.value == \"\")\n";
			echo "\t{\n";
			echo "\t\talert(\"请输入主题名称！\");\n";
			echo "\t\ttheForm.title.focus();\n";
			echo "\t\treturn (false);\n";
			echo "\t}\n";
			echo "\tif (theForm.content.value == \"\")\n";
			echo "\t{\n";
			echo "\t\talert(\"请输入主题内容！\");\n";
			echo "\t\ttheForm.content.focus();\n";
			echo "\t\treturn (false);\n";
			echo "\t}\n";
			echo "}\n";
			echo "</script>\n";
			echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
			echo "<tr>\n";
			echo "<td>";
			echo "<center><h2>编辑贴子</h2></center>";
			echo "</td>\n";
```

```
            echo "</tr>\n";
            echo "</table>\n";
            $sql_t="select p_id,topic_name from $table_topic where id='$rows[topic_id]'";
            $result_t=mysql_query($sql_t,$link) or die(mysql_error());
            $rows_t=mysql_fetch_array($result_t);
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<tr>\n";
            echo "<td><a href=20-5.php>论坛首页</a>|<a href=20-6.php?id=".$rows_t[0].">主论坛</a>|<a
href=20-7.php?id=".$rows[topic_id].">".$rows_t[1]."</a></td>\n";
            echo "</tr>\n";
            echo "</table>\n";
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<form method=\"post\" action=\"20-12.php?id=".$_GET[id]."\" onsubmit=\"return
juge(this)\">\n";
            echo "<tr>\n";
            if($rows[re_id]==0) $re_id=$_GET[id];                    //如果贴子是主题
            else $re_id=$rows[re_id];                                //如果贴子不是主题
            echo "<input type=\"hidden\" name=\"re_id\" value=\"".$re_id."\">\n";
            echo "<td>输入标题：</td>\n";
            echo "<td><input type=\"text\" name=\"title\" value=\"".$rows[title]."\"></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td>输入内容：</td>\n";
            echo "<td><textarea rows=5 cols=50 name=\"content\">".$rows[content]."</textarea></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td colspan=\"2\"><center><input type=submit value=提交><input type=reset value=重置
></center></td>\n";
            echo "</tr>\n";
            echo "<form>";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>\n";
            echo "</html>\n";
        }
        else                                                         //如果内容已经提交
        {
            $title=$_POST[title];
            $content=$_POST[content];
            $id=$_GET[id];
            $re_id=$_POST[re_id];
            $sql="update $table_posts set title='$title',content='$content' where id='$id'";
            if(mysql_query($sql,$link))                              //如果更新语句顺利执行
            {
                echo "编辑贴子成功，现在返回主题显示页";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-9.php?id=".$re_id."\">";
            }
            else
            {
```

```
                    echo "编辑记录时出错，现在返回";
                    echo "<meta http-equiv=\"refresh\" content=\"2; url=20-9.php?id=".$re_id."\">";
                }
            }
        }
}
?>
```

把以上代码保存为“20-12.php”。然后，如果想对哪一条记录进行编辑，就可以单击图 20.23 某一条贴子标题后面的“编辑” 链接即可。当然前提是要有权限，如果没有相应的权限，就不会显示出编辑链接。但是有可能有恶意的使用者通过猜想直接在地址栏中运行。所以还要在编辑页里进行用户身份的判断。如果是不合法用户请求编辑页，就会出现如图 20.24 所示的提示。

这里以管理员身份登录，编辑如图 20.23 所示的一条 TEST 记录。单击过链接后，出现如图 20.25 所示的执行结果。



图 20.24　非法用户请求编辑页执行结果



图 20.25　合法用户请求编辑页执行结果

在这里可以对贴子的标题及内容进行修改，修改完成后直接单击“提交”按钮即可。这里把贴子内容修改为与 PHP 编程有关系的内容，这样也符合 PHP 编程区内容的要求。提交操作，如果正确执行，程序将会返回到主题显示页。执行操作后的主题显示页如图 20.26 所示。



图 20.26　执行修改操作后的主题显示页

经过操作，贴子内容及标题已经修改成功。

20.7.2　删除贴子

单纯的编辑贴子，有时并不能满足要求，因为总是有别有用心的人会在论坛中放很多垃圾贴。有时是无用的恶意广告贴，有时有色情网站链接贴，有的则含有违犯国家法律、法规内容的贴子，对于这一类贴子必须给予删除。这一小节介绍如何删除贴子。

在进行删除操作之前同编辑贴子一样，也需要对论坛的主题显示页做一下改动。在 20.7.1 小节加入编辑链接的后面再跟上删除链接。

具体加入如下代码：

```
if($rows_t[admin]==3)
echo "<a href=\"20-13.php?id=".$rows[id]."\">删除</a>";
```

上面这段代码实现的功能是，先判断用户身份，如果用户是管理员则显示删除链接。这样就确保了只有管理员才能单击相应的链接。修改后主题的显示结果如图 20.27 所示。



图 20.27　加入删除链接的主题显示效果

准备工作做完，下面介绍如何对贴子进行删除操作。这里还要分两种情况，一种是被删除的是非主题。这种情况很好办，直接删除记录即可。另外一种情况是删除的记录是主题。这就比较麻烦了，因为主题通常都有回复，所以不光要对贴子记录本身进行操作，还要对主题的回复进行操作。

这里采用这样的处理机制，如果删除的贴子本身是主题就查找它的所有回复，把第 1 条对主题的回复变为主题，原来对原主题的回复都变成了对新主题的回复。听起来好像很复杂，实际上并不复杂。

如有一个主题，“1”。其下有 3 个回复“1-1”、“1-2”、“1-3”。则删除主题“1”，原来的回复“1-1”就成为了新的主题，而“1-2”、“1-3”则成了对新主题的回复。

下面给出具体的代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>删除贴子</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])
```

```
{
	echo "<h2>没有请求 ID</h2>";
	echo "<h3>点<a href=20-5.php>这里</a>返回</h3>";
	exit();
}
else
{
	require "20-1.php";
	$sql="select admin from $table_members where name='$_COOKIE[user]'";
	$result=mysql_query($sql,$link) or die(mysql_error());
	$rows=mysql_fetch_array($result);
	if($rows[0]<3)
	{
		echo "<h2>没有这样的权限</h2>";
		echo "<h3>点<a href=\"#\" onclick=\"history.go(-1)\">这里</a>返回</h3>";
		exit();
	}
	else
	{
		$sql="select re_id from $table_posts where id='$_GET[id]'";
		$result=mysql_query($sql,$link) or die(mysql_error());
		$rows=mysql_fetch_array($result);
		if($rows[re_id]==0)
		{
			$sql2="select id from $table_posts where re_id='$_GET[id]' limit 1";
			$result2=mysql_query($sql2,$link);
			$rows2=mysql_fetch_array($result2);
			$sql3="update $table_posts set re_id='0' where id='$rows2[0]'";
			mysql_query($sql3,$link);
			$sql4="update $table_posts set re_id='$rows2[0]' where re_id='$_GET[id]'";
			mysql_query($sql4,$link);
		}
		$sql="delete from $table_posts where id='$_GET[id]'";
		if(mysql_query($sql,$link))
		{
			echo "删除贴子成功，现在返回论坛首页";
		}
		else
		{
			echo "删除记录时出错，现在返回论坛首页";
		}
		echo "<meta http-equiv=\"refresh\" content=\"2; url=20-5.php\">";
	}
}
?>
```

把以上代码保存为“20-13.php”。单击图 20.27 中的“删除”链接，就相当于执行了 20-13.php。同理，这个页面也需要对用户的身份进行验证，必须是管理员才能进行这项操作。如果普通用户或者匿名用户执行这个页面，则会出现与图 20.24 类似的提示页。如果是有权限的管理员单击此链接，则会

执行相关的删除操作。删除主题后的主题页面如图 20.28 所示。

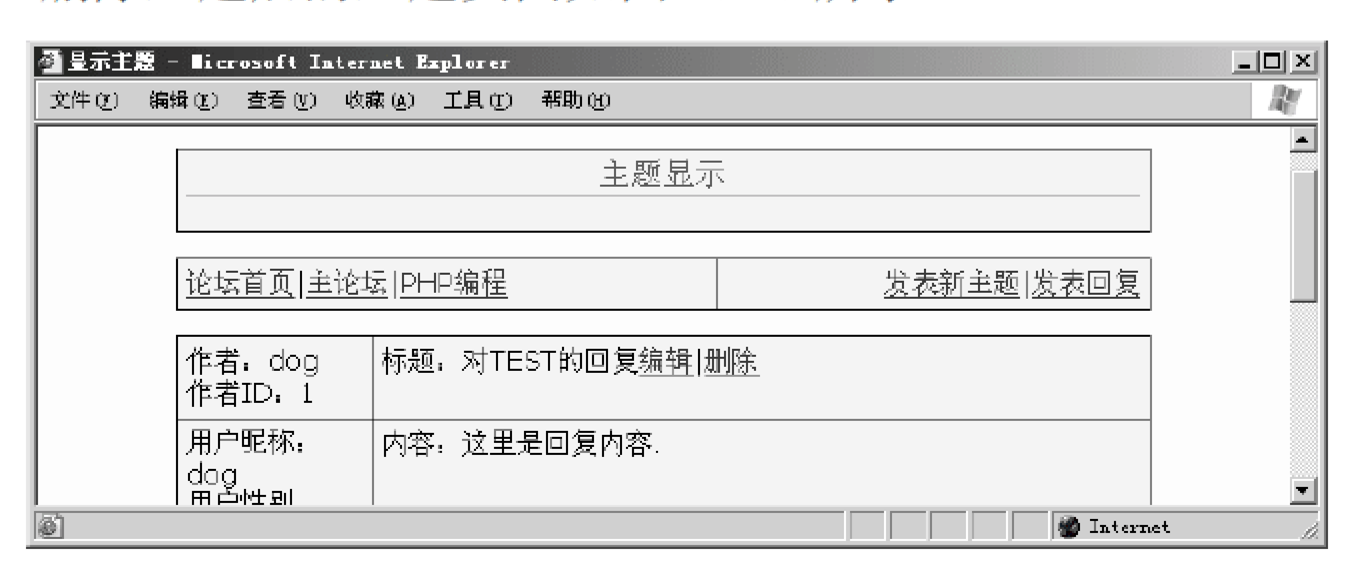


图 20.28　删除主题后的显示

可以看到，原来的回复变成了新的主题，说明删除操作成功。

20.7.3　防掘墓功能

什么是掘墓呢？在分论坛显示页中，显示的是各个主题的基本内容。排列顺序是按照回复的先后顺序进行排列的。这样可以使得用户最后回复的贴子始终处于最上方，使其他用户能够第一时间浏览到这个贴子及回复，如图 20.29 所示。

分论坛显示 - Microsoft Internet Explorer

分论坛首页——PHP编程

PHP编程专区，欢迎各位PHP菜鸟大虾光临！

论坛首页|主论坛　　发表新主题

标题	作者	查看	回复	最后发表
对TEST的回复	dog	2	0	2006年10月24日11：32：23
第一条测试信息	dog	31	0	2006年10月23日20：46：38

图 20.29　最后回复的贴子显示于最上方

这一设定本来是为了方便用户能够及时发现最新的贴子的，但这样就会存在一些问题，会有一些无聊的用户，对很久以前的老贴进行回复。这样一来，老主题就会显示在分论坛页面最醒目的位置。网络上通常对这种恶意回复老贴子的行为叫做掘墓。

如何防止掘墓行为的发生呢？比较可行的方法就是先设定一个最大允许回复天数。如这一天数设为“5”，表示如果主题留言时间是在 5 天以前，那么就不允许再对这一主题进行回复了。这样就可以有效避免掘墓行为的发生。

这要求在对贴子进行回复时先判断是否已经超过了最大允许回复天数。如果超过了，就给出不允许回复的提示。这就要对 20-8.php 进行修改，因为这个文件是发表主题或者回复时要用到的页面。

下面给出具体修改措施。

找到以下内容：

```
if(!$_GET[re_id]) $re_id=0;
else $re_id=$_GET[re_id];
```

在这些内容下面添加如下内容：

```
function next7($y,$m,$d)
{
    $t["01"]=$t["03"]=$t["05"]=$t["07"]=$t["08"]=$t["10"]=$t["12"]=31;
    $t["02"]=28;
    $t["04"]=$t["06"]=$t["09"]=$t["11"]=30;
    if($d+5<=$t[$m]) $d=$d+5;
    else
    {
        $d=$d+5-$t[$m];
        $m=$m+1;
        if(strlen($m)==1) $m="0".$m;
        if($m>12) $y=$y+1;
    }
    if(strlen($d)==1) $d="0".$d;
    return($y.$m.$d);
}
if($re_id!=0)
{
    $sql="select post_time from $table_posts where id='$re_id'";
    $result=mysql_query($sql);
    $rows=mysql_fetch_array($result);

$last=next7(substr($rows[post_time],0,4),substr($rows[post_time],6,2),substr($rows[post_time],10,2));
    $temp=date(Ymd);
    if($last<$temp)
    {
        echo "已经超过 5 天了，你还回复什么？！";
        echo "<br>点<a href=20-7.php?id=". $_GET[topic_id].">这里</a>返回";
        exit();
    }
}
```

这里首先定义了一个函数 next7()，这一函数的作用就是求某一天的 5 天后是哪一天。然后读取要回复的主题的日期，截取出整数，再求其 5 天后是多少，把值赋给$last 变量。然后再与代表当前日期的变量$temp 相比较，如果$last 小于$temp，则给出出错提示，并且中止所有操作。

经过这样一处理，用户的恶意掘墓行为就会得到有效的控制。

关于贴子的管理，这一节就介绍到这里。20.8 节介绍论坛用户的管理。

20.8　用 户 管 理

一个用户在注册一个论坛后，其各项信息，包括他的密码、昵称，甚至性别、电子信箱、图像、签名等内容都是可以改变的。通常这些操作也都是由用户自行完成。这一节就来讲如何实现用户的管理。

20.8.1　用户信息的显示

用户与用户之间都应该能够互相查看对方的注册信息。在显示主题时，在贴子的作者一项上加上指向该用户信息显示页的超链接，以实现不同用户间的互相查看。

这里就需要对 20-9.php 做一点修改。找到如下的内容：

```
echo "作者：".$rows[poster];
```

把它修改为：

```
echo "作者：<a href=20-14.php?id=".$rows[poster_id].">".$rows[poster]."</a>";
```

经过这样的修改，原本只显示的用户名，现在有了一个指向，即查看用户信息页面的超链接。

下面就如何显示用户信息为读者做介绍。获取参数$_GET[id]，然后从用户表中搜索相关记录，显示出相关信息即可。

具体内容请参看以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>显示用户信息</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])
{
    echo "<h2>没有请求 ID</h2>";
    echo "<h3>点<a href=# onclick=history.go(-1)>这里</a>返回</h3>";
    exit();
}
else
{
    require "20-1.php";
    $sql="select * from $table_members where id='$_GET[id]'";
    $result=mysql_query($sql,$link) or die(mysql_error());
    $rows=mysql_fetch_array($result);
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    echo "<tr>\n";
    echo "<td>";
    echo "<center><h2>显示用户".$rows[name]."信息</h2></center>";
    echo "</td>\n";
    echo "</tr>\n";
    echo "</table>\n";
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    echo "<tr>\n";
    echo "<td>";
```

```
echo "<a href=\"20-5.php\">论坛首页</a>";
echo "</td>\n";
if($rows[name]==$_COOKIE[user])
{
    echo "<td align=\"right\">";
    echo "<a href=\"20-15.php?id=".$rows[id]."\">编辑信息</a>|";
    echo "<a href=\"20-16.php?id=".$rows[id]."\">修改密码</a>";
    echo "</td>\n";
}
echo "</tr>\n";
echo "</table>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td>用户名：</td>\n";
echo "<td>".$rows[name]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户昵称：</td>\n";
echo "<td>".$rows[nickname]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>性别：</td>\n";
echo "<td>".$rows[sex]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>电子信箱：</td>\n";
echo "<td>".$rows[email]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户图像</td>\n";
echo "<td>";
if($rows[photo]) echo "<img src=images/".$rows[photo].">";
else echo "该用户没有设置图像";
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>签名：</td>\n";
echo "<td>";
if($rows[q_name]) echo $rows[q_name];
else echo "该用户没有设置签名档";
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>发贴数量：</td>\n";
echo "<td>".$rows[post_num]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>注册日期：</td>\n";
echo "<td>".$rows[reg_date]."</td>\n";
```

```
        echo "</tr>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "</html>\n";
    }
?>
```

把上面这段代码保存为“20-14.php”。在 PHP 环境下执行该文件，如果是匿名用户浏览某用户的信息，其执行结果会如图 20.30 所示。从图 20.30 的执行结果可以看到页面只显示出了用户的基本信息。如果某用户查看自己的信息，执行结果将会有如图 20.31 所示。







图 20.30　匿名用户浏览其他用户的信息　　　　图 20.31　注册用户查看自己的信息

可以看到，注册用户查看自己的信息，会显示出指向编辑信息与修改密码页面的链接。用户可以通过这些连接轻松实现对自己的信息及密码的修改。

20.8.2 小节介绍如何使用户能够修改自己的注册信息。

20.8.2　普通信息的修改

这一小节介绍如何使用户能够修改自己的普通信息。当然用户也不能随意修改自己的所有信息，如用户名、发贴数、注册日期、管理员等级就不能让用户随意修改。除此之外能够改动的就只有昵称、性别、电子信箱、论坛图像、发贴个性签名这些内容了。下面介绍如何实现对用户这些信息的修改。

具体内容请参看如下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改用户信息</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
```

```
if(!$_GET[id])
{
    echo "<h2>没有请求 ID</h2>";
    echo "<h3>点<a href=# onclick=history.go(-1)>这里</a>返回</h3>";
    exit();
}
else
{
    require "20-1.php";
    $sql="select * from $table_members where id='$_GET[id]'";
    $result=mysql_query($sql,$link) or die(mysql_error());
    $rows=mysql_fetch_array($result);
    if($rows[name]!=$_COOKIE[user])
    {
        echo "<h2>请求 ID 与用户身份不符！ </h2>";
        echo "<h3>点<a href=# onclick=history.go(-1)>这里</a>返回</h3>";
        exit();
    }
    else
    {
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<tr>\n";
        echo "<td>";
        echo "<center><h2>修改用户".$rows[name]."信息</h2></center>";
        echo "</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<tr>\n";
        echo "<td>";
        echo "<a href=\"20-5.php\">论坛首页</a>";
        echo "</td>\n";
        echo "<td align=\"right\">";
        echo "<a href=\"20-14.php?id=".$rows[id]."\">显示信息</a>|";
        echo "<a href=\"20-16.php?id=".$rows[id]."\">修改密码</a>";
        echo "</td>\n";
        echo "</tr>\n";
        echo "</table>\n";
        if(!$POST[$nickname])
        {
            echo "<script language=\"javascript\">\n";
            echo "function juge(theForm)\n";
            echo "{\n";
            echo "\tif (theForm.nickname.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入昵称！ \");\n";
            echo "\t\ttheForm.title.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
```

```
echo "}\n";
echo "function s_photo(the)\n";
echo "{\n";
echo "\tdocument.img.src='images/'+the.photo.value+'.bmp';\n";
echo "}\n";
echo "</script>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<form method=\"post\" action=\"20-15.php?id=".$_GET[id]."\" onsubmit=\"return juge(this)\">\n";
echo "<tr>\n";
echo "<td>用户昵称：</td>\n";
echo "<td>";
echo "<input type=\"text\" name=\"nickname\" value=\"".$rows[nickname]."\">";
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>性别：</td>\n";
echo "<td>";
echo "<input type=\"radio\" name=\"sex\" value=\"boy\" ";
if($rows[sex]=="boy") echo " checked ";
echo ">男";
echo "<input type=\"radio\" name=\"sex\" value=\"girl\" ";
if($rows[sex]=="girl") echo " checked ";
echo ">女";
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>电子信箱：</td>\n";
echo "<td>";
echo "<input type=\"text\" name \"email\" value=\"".$rows[email]."\">";
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户图像</td>\n";
echo "<td>";
echo "<select name=\"photo\" size=\"1\" onchange=\"s_photo(this.form)\">\n";
for($i=1;$i<21;$i++)
{
    echo "<option value=".$i.">".$i."</option>\n";;
}
echo "</select>";
if(!$rows[photo])
{
    echo "用户没有设置图像<br>";
    echo "<img src=\"images/1.bmp\" name=\"img\">";
}
else
{
    echo "<img src=\"images/".$rows[photo]."\" name=\"img\">";
```

```
                    }
                    echo "</td>\n";
                    echo "</tr>\n";
                    echo "<tr>\n";
                    echo "<td>签名：</td>\n";
                    echo "<td>";
                    echo "<input type=\"text\" name \"q_name\" value=\"".$rows[q_name]."\">";
                    echo "</td>\n";
                    echo "</tr>\n";
                    echo "<tr>\n";
                    echo "<td colspan=\"2\"><center>";
                    echo "<input type=submit value=\"提交\">";
                    echo "</center></td>\n";
                    echo "</tr>\n";
                    echo "</form>";
                    echo "</table>\n";
                    echo "</center>\n";
                    echo "</body>\n";
                    echo "</html>\n";
                }
                else
                {
                    $nickname=$_POST[nickname];
                    $sex=$_POST[sex];
                    $email=$_POST[email];
                    $photo=$_POST[photo];
                    $q_name=$_POST[q_name];
                    $id=$_GET[id];
                    $sql="update                    $table_members                    set
nickname='$nickname',sex='$sex',email='$email',photo='$photo',q_name='$q_name' where id='$id'";
                    if(mysql_query($sql,$link))
                    {
                        echo "修改注册信息成功，现在返回查看信息页";
                        echo "<meta http-equiv=\"refresh\" content=\"2; url=20-14.php?id=".$id."\">";
                    }
                    else
                    {
                        echo "修改注册信息失败，现在返回修改用户记录页";
                        echo "<meta http-equiv=\"refresh\" content=\"2; url=20-15.php?id=".$id."\">";
                    }
                }
            }
        }
        ?>
```

把以上代码保存为“20-15.php”。单击图 20.31 中的“编辑信息”超链接，就可以打开该文件。打开后的执行结果如图 20.32 所示。

这里只对需要改动的内容进行更改，然后单击“提交”按钮，就可以把修改信息提交到后台进行处理。如果操作正确执行，新的用户将代替以前的信息。用户信息修改成功。



图 20.32　修改用户信息执行结果

20.8.3　用户密码的修改

密码是一个比较特殊的用户信息，它是用户登录系统的最重要的“钥匙”。所以要把对密码的修改与普通信息的修改分开来进行。

密码修改的机理是这样，先让用户分别输入旧密码及新的密码。然后在后台进行判断，如果用户所输入的旧密码与库里保存的一致，则执行把旧密码更改为新密码的 SQL 语句。反之，如果用户输入不一致，则给出出错提示。

下面给出具体代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改用户密码</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
if(!$_GET[id])
{
	echo "<h2>没有请求 ID</h2>";
	echo "<h3>点<a href=# onclick=history.go(-1)>这里</a>返回</h3>";
	exit();
}
else
{
	require "20-1.php";
	$sql="select * from $table_members where id='$_GET[id]'";
	$result=mysql_query($sql,$link) or die(mysql_error());
	$rows=mysql_fetch_array($result);
```

```
if($rows[name]!=$_COOKIE[user])
{
    echo "<h2>请求 ID 与用户身份不符！ </h2>";
    echo "<h3>点<a href=# onclick=history.go(-1)>这里</a>返回</h3>";
    exit();
}
else
{
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    echo "<tr>\n";
    echo "<td>";
    echo "<center><h2>修改用户".$rows[name]."的密码</h2></center>";
    echo "</td>\n";
    echo "</tr>\n";
    echo "</table>\n";
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    echo "<tr>\n";
    echo "<td>";
    echo "<a href=\"20-5.php\">论坛首页</a>";
    echo "</td>\n";
    echo "<td align=\"right\">";
    echo "<a href=\"20-14.php?id=".$rows[id]."\">显示信息</a>|";
    echo "<a href=\"20-15.php?id=".$rows[id]."\">修改信息</a>";
    echo "</td>\n";
    echo "</tr>\n";
    echo "</table>\n";
    if(!$_POST[password])
    {
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.password.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入旧密码！ \");\n";
        echo "\t\ttheForm.password.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.newpassword.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入新密码！ \");\n";
        echo "\t\ttheForm.newpassword.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.newpassword.value.length <8)\n";
        echo "\t{\n";
        echo "\t\talert(\"密码要在 8 位以上！ \");\n";
        echo "\t\ttheForm.newpassword.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.newpassword.value != theForm.repassword.value)\n";
```

```
			echo "\t{\n";
			echo "\t\talert(\"重复输入的密码不一致！\");\n";
			echo "\t\ttheForm.repassword.focus();\n";
			echo "\t\treturn (false);\n";
			echo "\t}\n";
			echo "}\n";
			echo "</script>\n";
			echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
			echo "<form method=\"post\" action=\"20-16.php?id=".$_GET[id]."\" onsubmit=\"return 
juge(this)\">\n";
			echo "<tr>\n";
			echo "<td>输入旧密码：</td>\n";
			echo "<td>";
			echo "<input type=\"password\" name=\"password\">";
			echo "</td>\n";
			echo "</tr>\n";
			echo "<tr>\n";
			echo "<td>输入新密码：</td>\n";
			echo "<td>";
			echo "<input type=\"password\" name=\"newpassword\">";
			echo "</td>\n";
			echo "</tr>\n";
			echo "<tr>\n";
			echo "<td>再输入一次：</td>\n";
			echo "<td>";
			echo "<input type=\"password\" name=\"repassword\">";
			echo "</td>\n";
			echo "</tr>\n";
			echo "<tr>\n";
			echo "<td colspan=\"2\"><center>";
			echo "<input type=submit value=\"提交\">";
			echo "</center></td>\n";
			echo "</tr>\n";
			echo "</form>";
			echo "</table>\n";
			echo "</center>\n";
			echo "</body>\n";
			echo "</html>\n";
		}
		else
		{
			$password=md5($_POST[password]);
			$newpassword=md5($_POST[newpassword]);
			$id=$_GET[id];
			$sql="select id from $table_members where name='$_COOKIE[user]' and 
password='$password'";
			$result=mysql_query($sql,$link);
			$nums=mysql_num_rows($result);
			if($nums<1)
			{
```

```
				echo "输入的用户密码错误！<p>";
				echo "请重新输入！";
				echo "<meta http-equiv=\"refresh\" content=\"2; url=20-16.php?id=".$id."\">";
			}
			else
			{
				$sql="update $table_members set password='$newpassword' where id='$id'";
				if(mysql_query($sql,$link))
				{
					echo "修改用户密码成功，现在返回查看信息页";
					echo "<meta http-equiv=\"refresh\" content=\"2; url=20-14.php?id=".$id."\">";
				}
				else
				{
					echo "修改用户密码失败，现在返回修改用户记录页";
					echo "<meta http-equiv=\"refresh\" content=\"2; url=20-15.php?id=".$id."\">";
				}
			}
		}
	}
}
?>
```

将上面代码保存为“20-16.php”。然后单击图 20.32 中的“修改密码”超链接，就可以执行该文件。其执行结果如图 20.33 所示。



图 20.33　用户修改密码执行结果

依次输入用户的旧密码、新密码、重复新密码，然后单击“提交”按钮，就可以把输入内容提交到后台进行处理。

如果用户输入正确，并且执行了相关操作，则用户的旧密码就会被新密码替代，以完成修改密码的操作。

20.8.4　用户退出页面

通常情况下，还应该设计一个用户退出登录的页面，以方便用户随时退出系统。为了使主程序能

够访问该页面，还应该对论坛首页文件进行修改，以加入该页的链接。

在 20-5.php 中找到以下内容：

```
echo "<td>";
if(!$_COOKIE[user])                    //如果没有用户登录，则显示登录连接
{
    echo "<a href=\"20-4.php\">用户登录</a>";
}
else                                   //如果用户已经登录，则显示登录用户名
{
    require "20-1.php";
    $sql="select id from $table_members where name='$_COOKIE[user]'";
    $result=mysql_query($sql,$link);
    $rows=mysql_fetch_array($result);
    echo "登录用户：<a href=20-14.php?id=".$rows[0].">".$_COOKIE[user]."</a>";
}
echo "</td>\n";
```

将其修改为：

```
echo "<td>";
if(!$_COOKIE[user])                    //如果没有用户登录，则显示登录连接
{
    echo "<a href=\"20-4.php\">用户登录</a>";
}
else                                   //如果用户已经登录，则显示登录用户名
{
    require "20-1.php";
    $sql="select id from $table_members where name='$_COOKIE[user]'";
    $result=mysql_query($sql,$link);
    $rows=mysql_fetch_array($result);
    echo "登录用户：<a href=20-14.php?id=".$rows[0].">".$_COOKIE[user]."</a>|";
    echo "<a href=20-17.php>退出登录</a>";
}
echo "</td>\n";
```

经过这样的修改就为主页面加入了用户退出的链接。

其实用户的退出操作极其简单，清空用户注册的 Cookie 记录，然后再跳转到指定页即可。下面给出具体实现代码：

```
<?php
setcookie("user","");
echo "<html>\n";
echo "<head>\n";
echo "<title>登录用户安全退出</title>\n";
echo "<meta http-equiv=\"refresh\" content=\"2; url=20-5.php\">";
echo "</head>\n";
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
echo "<h2>登录用户已经退出</h2>\n";
```

```
echo "<h3>两秒后返回论坛首页</h3>\n";
echo "</center>\n";
echo "</body>\n";
echo "</html>\n";
?>
```

把上面代码保存为“20-17.php”。当登录用户单击“退出登录” 链接后，用户就能安全地从系统退出。

至此，用户管理工作基本上就算是完整了，不但能够显示用户信息，而且能够更改用户信息，可以满足一般的要求了。

20.9　进一步完善

现在，论坛程序已经基本上能够满足大部分需求了，但与通常的论坛相比还存在很多不足。大致有以下几个方面：

- 不具有文件上传功能。通常的论坛都会在用户发表新主题或者回复时有一个文件上传界面允许用户上传指定类型及限制大小的文件。这个问题也很容易实现，关于如何上传文件请参看本书第 8 章的内容，这里只给出思路。当用户上传文件后，在提交用户发表主题或回复的内容最后加上指向文件下载页面的链接。而专门的文件下载页面则指向该上传的文件。这样就可以显示指定文件，用户也可以下载该文件了。
- 用户与用户之间不能传送信息。通常的论坛还会有一个信息系统，即不同的用户之间可以互相发送信息，只有接收人才能够查看。当用户登录论坛时，如果有未阅读的新信息，则系统会提示用户进行阅读。这也是一个相当有用的功能，实现起来也并不复杂。只需要在原有的表结构上再加一个信息表（MSN）。表一共有这样几个字段：ID、发件人、收件人、内容、是否显示、是否阅读等。然后为收、发信息建立专门的页面。任何用户都只能查看收件人为自己名字的信息。也可以给其他人发送信息。信息的阅读初始状态为 False，当用户登录论坛，用检查是否有收件人为用户名，阅读状态为 False 的信息，如果有就提醒用户进行阅读。信息在用户阅读过后，阅读状态就为 True。用户也可以选择删除自己收件箱中的信息，而实际上的删除操作只是把显示状态改为 False。这样就不再显示该信息。这里只给出了实现机制，有兴趣的读者可以自行研究以实现该功能。
- 没有为用户定义级别，普通用户无法体会到升级的乐趣。通常的论坛都会为用户根据发贴数量，或者积分设定级别，如 0~50 为初级用户，50~100 为中级用户，100~200 为高级用户，200~500 为超级用户，500~1000 为帝王等。这样可以鼓励用户不断发贴以提高自己的级别。要实现这样的功能有多种方法，一种就是把相应的级别直接写入用户表中的某个字段，如 LEVEL。在用户发过贴时判断其发贴数属于哪一个级别，再把相应级别写入该字段。这样随着发贴数的增加，用户的级别就会不断增加。但这样存在一个弊端，就是当用户级别固定后，想再更改就相当困难了，如以前的级别这样定义：初级、中级、高级、超级等，现在想把用户级别表示方式改为：学前班、小学生、初中生、高中生、学士、硕士、博士等。如果一条一条改就

会相当困难，而且在实际中根本是不可能的，所以可以采用这样的方法，不设定用户的级别字段。当在需要显示用户级别时，读取用户的发贴数与特定配置文件中的分数段表示值进行比较。用户属于哪个范围，就显示相应的级别，如现在有配置文件，定义 0~50 为初级用户，50~100 为中级用户，100~200 为高级用户，200~500 为超级用户，500~1000 为帝王。显示时把用户发贴数与这些配置进行比较，属于哪个范围，就显示相应的级别。而想更改级别表示方法时只需要更改配置文件即可。这样可以保证级别表示方法的可移植性。限于篇幅也不再给出具体代码，有兴趣的读者，可以去尝试进行。

- 只有一个超级管理员，各个分论坛没有相应的版主。如果要求版主，就要给每个论坛分类。不管是主分类还是二级分类都增加一个字段用于存放版主的 ID，然后再把相应用户的 ADMIN 值改为 1，以区别于普通用户，然后再给这些版主赋予相应权限即可。
- 对用户的输入没有过滤功能。用户的恶意输入会对论坛造成破坏，所以应该屏蔽用户的输入，以处理掉 HTML 标记。只需要使用 htmlspecialchars()函数对用户输入进行处理即可，这里也不再赘述。
- 其他更为高级的功能。一个完整的论坛还应有更多的功能，如在线用户列表、用户生日友情提示、版主可以奖励用户积分、置顶贴、精华贴等。可以这样说，要想讲清楚论坛的所有功能都需要一本书。由于本书的篇幅所限，只能讲些最基本的功能，更多的功能都有待读者在以后的学习中进一步研究完成。

20.10　本 章 小 结

这一章向读者介绍了如何用所学的 PHP 知识来实现一个简单的论坛。本章把一个论坛分解为表的设计、所需库表的创建、用户的注册与登录、论坛的显示、主题的发表与回复、论坛分类的管理、贴子的管理、用户的管理几大项，分别进行了说明。通过本章的学习，用户不仅进一步深化了 PHP 对 MySQL 数据库的操作。而且对于如何通过 PHP 代码、算法来解决实际问题都会有一个深刻的认识。

第21章

网上商城全站系统

- 网上商城系统分析
- 大型程序的数据库表设计
- 用户注册和登录
- 购物车的实现
- 前台显示
- 后台管理

这一章，结合前面所学的内容来制作一个网上商城全站系统。相信读者在上网冲浪时都见过各式各样的网上商城，注册用户可以在上面购买自己喜欢的商品，并向读者介绍如何使用 PHP 再加上 MySQL 数据库来实现一个网上商城全站系统。通过本章的学习，相信读者会对 PHP 及 MySQL 的知识有一个全新的认识。

21.1　系 统 分 析

在制作系统之前，首先需要分析系统所要实现的功能，以明确制作目的。只有目的明确才能有的放矢，使接下来的的工作事半功倍，提高工作效率。作为一个网上商城，面对的是用户，所以必不可少的就要有一个用户注册与登录系统，这是构建用户系统的前提。用户有系统管理员与普通用户之分。

1. 系统管理员

管理员在登录系统后，可以在后台对图书记录表进行增加和删除操作。他也可以在后台修改图书的类别，查看、操作回复用户的购书订单。

2. 普通用户

普通用户登录后，则能进行浏览图书、查找图书等操作。浏览图书可以给予用户一定的自由，如按种类浏览或者按添加时间浏览等。搜索图书也有一定的自由性，用户可以把图书名称、图书作者、出版图书的出版社、图书所属的类别或者价格等多种条件作为搜索条件对图书进行搜索。

普通用户也可以查看图书详细情况，并把图书加入购物车。添加到购物车后，用户就可以随时查看购物车内容，更改购物车。例如，可以更改购物车中某一类商品的数量，或删除某一类商品。在确定了所购图书后，在前台提交购物车，给商城下订单。

21.2　设计数据库表结构

数据表设计的成功与否直接影响到程序的执行效率。这一节完成数据库表结构的设计。在整个系统中，要实现系统分析所要求的功能，共需要建立 5 个表，即用户表、图书类型表、图书记录表、订单表和销售记录表等。

21.2.1　用户表的设计

用户表主要存放注册用户的信息。该表主要包括索引 ID、用户名、用户密码、用户图像、用户的地址、电子信箱、QQ 号码、MSN 地址、注册时间、购物数量等基本信息，其中还应有一字段用以判断用户身份，如果该字段值为“0”，则为普通用户；如果为为“1”，则表示用户为管理员。该表具体字段如表 21.1 所示。

表 21.1　用户表 user

字　段　名	数 据 类 型	是否允许为空	说　　明
id	int(5)	否	索引ID（主键、自动增1）
name	varchar(12)	否	用户名
password	varchar(40)	否	用户密码
photo	varchar(80)	否	用户图像

续表

字　段　名	数 据 类 型	是否允许为空	说　　明
address	varchar(80)	否	用户的地址
email	varchar(80)	否	电子信箱
qq	varchar(15)	否	QQ号码
msn	varchar(80)	否	MSN地址
reg_date	varchar(20)	否	注册时间
post_num	int(5)	否	发贴数量
admin	int(1)	否	用户身份判断

21.2.2　图书类型表的设计

图书类型表存放图书的类别，图书的类别有二级分类，如主类别为电脑类用书，二级类别可以分别为硬件类、软件类、网络类、电子娱乐类等。要实现这样的分类，必须有图书类别表来存放。该表应该包含的基本信息有：索引 ID、主类别 ID、类别名称、类别介绍、类别数量等。该表具体字段如表 21.2 所示。

表 21.2　类型表 type

字　段　名	数 据 类 型	是否允许为空	说　　明
id	int(5)	否	索引ID（主键、自动增1）
p_id	int(5)	否	子类别所属主类别ID
type_name	varchar(12)	否	类别名称
type_description	varchar(80)	否	类别描述
type_num	int(5)	否	本类别的贴子数，当发本类贴时，该值增加1

21.2.3　图书记录表的设计

图书记录表存入图书的所有主要信息，大致有以下内容：索引 ID、书名、作者、出版社、所属子分类、价格、内容简介、封面扫描图、已售出量和存货量等。该表具体字段如表 21.3 所示。

表 21.3　图书记录表 book

字　段　名	数 据 类 型	是否允许为空	说　　明
id	int(5)	否	索引ID（主键、自动增1）
book_name	varchar(40)	否	图书的名称
book_author	varchar(20)	否	图书的作者
book_pub	varchar(40)	否	图书的出版社
book_type	int(5)	否	图书所属类别
book_cost	varchar(6)	否	图书售价
book_description	varchar(200)	否	图书描述
book_photo	varchar(80)	否	图书封面扫描图

续表

字　段　名	数 据 类 型	是否允许为空	说　　明
book_sale_num	int(5)	否	图书售出量
book_num	int(5)	否	图书存货量

21.2.4　订单记录表的设计

订单是这样一个概念，用户在选择了相应商品并提交后，提交的这些记录将以订单的形式保存在订单记录表中。只有当管理者回复了这些订单，并且完成了交易，相应的订单才会变成真正的销售记录。一个订单表应该包括如下内容：索引 ID、提交者 ID、提交者用户名、所购图书信息（这其中包括所购图书 ID、数量）、订单总额（为此次所有购书的总额）、回复状态（用户提交后，初始状态为 False，当管理员回复该订单后，状态变为 True）、提交时间（用户提交订单的时间）等。该表具体字段如表 21.4 所示。

表 21.4　订单记录表 order

字　段　名	数 据 类 型	是否允许为空	说　　明
id	int(5)	否	索引ID（主键、自动增1）
order_user_id	int(5)	否	提交者名称
order_user_name	varchar(20)	否	提交者ID
order_book_id	int(5)	否	所购图书ID
order_book_num	int(5)	否	购书数量
order_content	varchar(80)	否	订单内容
order_cost	varchar(10)	否	订购价格
order_state	enum('true','false')	否/默认值为false	订单状态
order_date	varchar(40)	否	订购时间

21.2.5　销售记录表的设计

销售记录表是直接由订单表变化而来的，用户提交的订单经管理员回复后就变成实实在在的销售记录，所以，该表内容与订单表基本类似，只不过多一个回复时间字段。该表具体字段如表 21.5 所示。

表 21.5　用户表 user

字　段　名	数 据 类 型	是否允许为空	说　　明
id	int(5)	否	索引ID（主键、自动增1）
sale_order_id	int(5)	否	对应订单ID
sale_date	varchar(40)	否	销售时间（即管理员回复时间）

至此，该系统所需要的 5 个表全部设计完毕。从 21.3 节开始将通过具体的代码来实现网上商城全站系统的所有功能。

21.3　准 备 工 作

在开始实现网上商城全站系统的所有功能之前，要做一些必要的准备工作。其中一项就是建立配置文件，以便以在后面的程序中调用。另一个是要创建系统运行所需要的各种表，即进行系统的安装。这一节将具体实现这些工作。

21.3.1　配置文件的创建

和前面几章所讲到与数据库相关的程序一样，程序所访问的主机名、连接主机的用户名、用户密码及数据库名通常都是固定不变的。各种表名，在创建后也是不会改变的，所以应该把这些重要的变量单独拿出来做成配置文件，以便其他程序文件来调用。在配置文件中存放着程序运行所需要的数据库的主机名、连接主机的用户名、密码、数据库名和表名等信息。

下面就来建立配置文件。具体内容参看以下代码：

```
<?
$db_host="localhost";                                    //主机名
$db_user="root";                                         //用户名
$db_pass="";                                             //密码
$db_name="test";                                         //数据库名
$table_user="user";                                      //用户表
$table_type="type";                                      //图书分类表
$table_book="book";                                      //图书记录表
$table_order="order";                                    //订单表
$table_sale="sale";                                      //销售记录表
$link=mysql_connect($db_host,$db_user,$db_pass);         //连接主机
mysql_select_db($db_name,$link);                         //选择数据库
?>
```

将以上代码保存为“21-1.php”，以便在以后的具体代码中调用该配置文件。

21.3.2　安装文件的创建

创建程序运行所需要的各种表是程序运行的前提，这一小节就来实现表的创建。以下代码实现的功能是先让用户输入各种内容包括注册的管理员信息、表的前缀等信息。然后，在后台获取用户提交的内容，创建各种表，并为用户表插入管理员的信息。

下面给出详细代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>安装程序</title>\n";
echo "</head>\n";
```

```
echo "<body>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
if(!$_POST[admin])                                    //如果没有默认参数，则显示 HTML
{
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.admin.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入管理员名称！ \");\n";
        echo "\t\ttheForm.admin.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pass.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入管理员密码！ \");\n";
        echo "\t\ttheForm.pass.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pass.value.length < 8 )\n";
        echo "\t{\n";
        echo "\t\talert(\"密码至少要 8 位！ \");\n";
        echo "\t\ttheForm.pass.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.re_pass.value !=theForm.pass.value)\n";
        echo "\t{\n";
        echo "\t\talert(\"确认密码与密码不一致！ \");\n";
        echo "\t\ttheForm.re_pass.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pre.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入表前缀！ \");\n";
        echo "\t\ttheForm.pre.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "function s_photo(the)\n";
        echo "{\n";
        echo "\tdocument.img.src='images/'+the.photo.value+'.bmp';\n";
        echo "}\n";
        echo "</script>\n";
        echo "<center>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">安装程序</font></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
```

```
        echo "<td>管理员：（后台登录）</td>\n";
        echo "<td><input type=\"text\" name=\"admin\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>管理员密码：（不小于 8 位）</td>\n";
        echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>确认密码：</td>\n";
        echo "<td><input type=\"password\" name=\"re_pass\" size=\"21\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>管理员 E-mail：（可选）</td>\n";
        echo "<td><input type=\"text\" name=\"email\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择图像：</td>\n";
        echo "<td>";
        echo "<select name=\"photo\" size=\"1\" onchange=\"s_photo(this.form)\">\n";
        for($i=1;$i<21;$i++)
        {
            echo "<option value=".$i.">".$i."</option>\n";;
        }
        echo "</select>";
        echo "<img src=\"images/1.bmp\" name=\"img\">";
        echo"</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>表的前缀：</td>\n";
        echo "<td><input type=\"text\" name=\"pre\" value=\"book_\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>\n";
        echo "<input type=\"submit\" value=\"下一步\">\n";
        echo "<input type=\"reset\" value=\"重新填\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
}
else                                        //如果有 POST 参数，则执行操作
{
        $name=$_POST[admin];                //获得参数
        $password=md5($_POST[pass]);        //获得密码，并使用 MD5 进行加密操作
        $email=$_POST[email];
        $photo=$_POST[photo];
        $pre=$_POST[pre];
```

```
require "21-1.php";
$table_user=$pre.$table_user;
$table_type=$pre.$table_type;
$table_book=$pre.$table_book;
$table_order=$pre.$table_order;
$table_sale=$pre.$table_sale;
$time=date("Y 年 m 月 d 日");
$sql="create table $table_user(
id int(5) not null auto_increment primary key,
name varchar(12) not null,
password varchar(40) not null,
photo varchar(80) not null,
address varchar(80) not null,
email varchar(80) not null,
qq varchar(15) not null,
msn varchar(80) not null,
reg_date varchar(20) not null,
post_num int(5) not null default 0,
admin int(1) not null
)";
mysql_query($sql,$link) or die(mysql_error());   //发送创建 user 表的 SQL 请求
$sql="create table $table_type(
id int(5) not null auto_increment primary key,
p_id int(5) not null,
type_name varchar(12) not null,
type_description varchar(80) not null,
type_num int(5) not null default 0
)";
mysql_query($sql,$link) or die(mysql_error());   //发送创建 type 表的 SQL 请求
$sql="create table $table_book(
id int(5) not null auto_increment primary key,
book_name varchar(40) not null,
book_author varchar(20) not null,
book_pub varchar(40) not null,
book_type int(5) not null,
book_cost varchar(6) not null,
book_description varchar(200) not null,
book_photo varchar(80) not null,
book_sale_num int(5) not null,
book_num int(5) not null
)";
mysql_query($sql,$link) or die(mysql_error());   //发送创建 book 表的 SQL 请求
$sql="create table $table_order(
id int(5) not null auto_increment primary key,
order_user_id int(5) not null,
order_user_name varchar(12) not null,
order_book_id int(5) not null,
order_book_num int(5) not null,
order_content varchar(80) not null,
order_cost varchar(10) not null,
```

```
order_state enum('true','false') not null default 'false',
order_date varchar(40) not null
)";
mysql_query($sql,$link) or die(mysql_error());  //发送创建 order 表的 SQL 请求
$sql="create table $table_sale(
id int(5) not null auto_increment primary key,
sale_order_id int(5) not null,
sale_date varchar(40) not null
)";
mysql_query($sql,$link) or die(mysql_error());  //发送创建 sale 表的 SQL 请求
$sql="insert into $table_type(p_id,type_name,type_description)values('0','主类别 1','系统创建的默认主类别
')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加默认主分类的 SQL 请求
$sql="insert into $table_type(p_id,type_name,type_description)values('1','分类别 1','系统创建的默认分类别
')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加默认子分类的 SQL 请求
$sql="insert                                                          into
$table_user(name,password,photo,address,email,qq,msn,reg_date,admin)values('$name','$password','$photo',
'','$email','','','$time','3')";
mysql_query($sql,$link) or die(mysql_error());  //发送添加管理员信息的 SQL 请求
$fp=fopen("21-1.php","w+");                     //将更新过的数据写入配置文件
fputs($fp,"<?\n");
fputs($fp,"\$db_host=\"localhost\";\n");
fputs($fp,"\$db_user=\"root\";\n");
fputs($fp,"\$db_pass=\"\";\n");
fputs($fp,"\$db_name=\"test\";\n");
fputs($fp,"\$table_user=\"$table_user\";\n");
fputs($fp,"\$table_type=\"$table_type\";\n");
fputs($fp,"\$table_book=\"$table_book\";\n");
fputs($fp,"\$table_order=\"$table_order\";\n");
fputs($fp,"\$table_sale=\"$table_sale\";\n");
fputs($fp,"\$link=mysql_connect(\$db_host,\$db_user,\$db_pass);\n");
fputs($fp,"mysql_select_db(\$db_name,\$link);\n");
fputs($fp,"?>");
fclose($fp);
echo "<center>";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td align=\"center\"><font size=\"5px\">安装程序</font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td align=\"center\"><font size=\"3px\">成功安装！ </font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td align=\"center\"><font size=\"3px\">删除该文件，以减少潜在危险！ </font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td align=\"center\">点<a href=\"21-5.php\">这里</a>进入</td>\n";
echo "</tr>\n";
echo "</table>\n";
```

```
        echo "</center>";
        echo "</body>\n";
        echo "</html>\n";
    }
    ?>
```

将上面所显示的代码保存为“21-2.php”。然后，在 PHP 执行环境下执行该文件，完成整个系统运行所需要的各种表的创建。该文件执行的前台结果如图 21.1 所示。

安装程序 - Microsoft Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
安装程序
管理员：（后台登录）
管理员密码：（不小于8位）
确认密码：
管理员E-mail：（可选）
选择图像：
表的前缀： book_
下一步 重新填
完毕 Internet

图 21.1　首次运行安装文件的执行结果

按照要求填入全部信息，全部填写完毕单击“下一步”按钮，将出现如图 21.2 所示的执行结果。

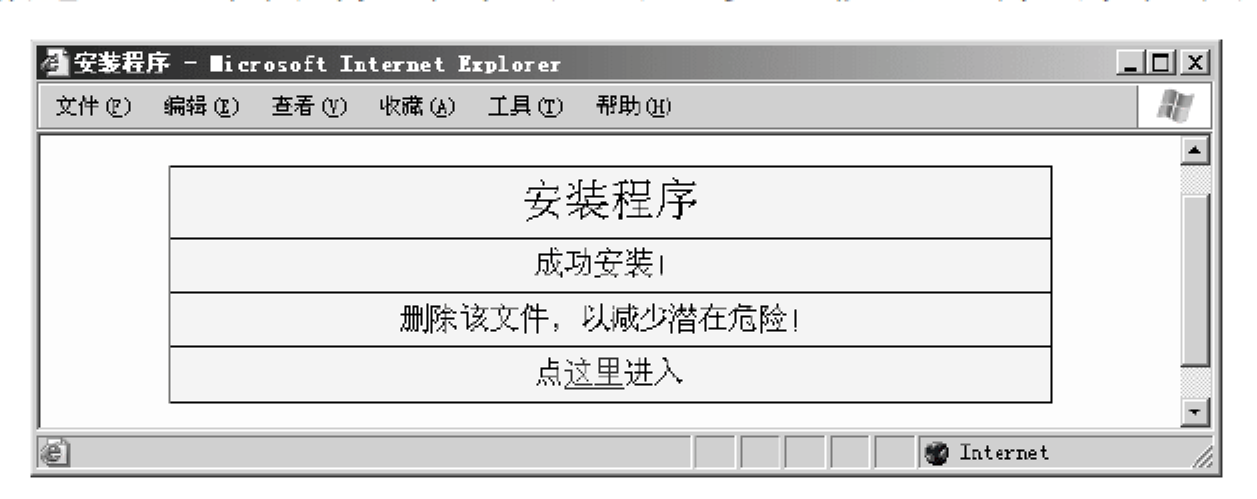


图 21.2　安装完成执行结果

成功执行安装后，系统运行所需要的表创建完毕。

随着安装程序正确执行完安装过程，系统实现前的准备工作就告一段落。从 21.4 节开始就将通过具体的代码来实现所需的各项功能。

21.3.3　头文件的创建

这一节来创建一个头文件，它可以方便的链接到其他页面。在其他文件中调用该文件，一方面可以减少代码量，另一方面也可以使各页面样式保持一致。由于该文件输出内容都是最普通的 HTML 内容，所以不过多解释。

下面给出具体代码：

```
<?php
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\" align=\"center\"     bgcolor=\"#000000\">\n";
echo "<tr>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"21-5.php\">商城首页</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"21-6.php\">所有图书</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"21-7.php\">分类查看</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"21-6.php\">图书搜索</a></center></td>\n";
echo "<td bgcolor=\"#cccc99\"><center><a href=\"21-8.php\">管理入口</a></center></td>\n";
echo "</tr>\n";
echo "</table>\n";
echo "<p>";
?>
```

把以上代码保存为“21-3.php”，以方便其他页面的调用。

21.4　用户的注册与登录

网上商城也需要有用户才行，用户是各类商品的潜在购买者。所以，作为一个网上商城系统必须有一个配套的用户注册与登录系统。这一节就来完成用户的注册与登录功能。

21.4.1　用户注册

用户注册页的作用是给出一个用户界面，要求用户输入新的用户名及其他相关信息。然后与库中所保存的用户进行比较，如果用户名已经存在，则给出出错提示；反之，就提取用户的输入信息作为一条新的记录写入到表中。

下面给出具体代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>注册新用户</title>\n";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
if(!$_POST[user])                              //如果没有默认参数，则显示 HTML
{
     echo "<script language=\"javascript\">\n";
     echo "function juge(theForm)\n";
     echo "{\n";
     echo "\tif (theForm.user.value == \"\")\n";
     echo "\t{\n";
     echo "\t\talert(\"请输入注册用户名！\");\n";
     echo "\t\ttheForm.user.focus();\n";
```

```
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.pass.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入用户密码！\");\n";
echo "\t\ttheForm.pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.pass.value.length < 8 )\n";
echo "\t{\n";
echo "\t\talert(\"密码至少要 8 位！\");\n";
echo "\t\ttheForm.pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.re_pass.value !=theForm.pass.value)\n";
echo "\t{\n";
echo "\t\talert(\"确认密码与密码不一致！\");\n";
echo "\t\ttheForm.re_pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.address.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入用户地址！\");\n";
echo "\t\ttheForm.address.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "function s_photo(the)\n";
echo "{\n";
echo "\tdocument.img.src='images/'+the.photo.value+'.bmp';\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
echo "<tr>\n";
echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">注册新用户</font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户名：（后台登录）</td>\n";
echo "<td><input type=\"text\" name=\"user\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户密码：（不小于 8 位）</td>\n";
echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>确认密码：</td>\n";
echo "<td><input type=\"password\" name=\"re_pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
```

```
        echo "<td>用户 E-mail：（可选）</td>\n";
        echo "<td><input type=\"text\" name=\"email\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>ＱＱ号：（可选）</td>\n";
        echo "<td><input type=\"text\" name=\"qq\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>ＭＳＮ号：（可选）</td>\n";
        echo "<td><input type=\"text\" name=\"msn\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择图像：</td>\n";
        echo "<td>";
        echo "<select name=\"photo\" size=\"1\" onchange=\"s_photo(this.form)\">\n";
        for($i=1;$i<21;$i++)
        {
            echo "<option value=".$i.">".$i."</option>\n";;
        }
        echo "</select>";
        echo "<img src=\"images/1.bmp\" name=\"img\">";
        echo"</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>用户地址：</td>\n";
        echo "<td><input type=\"text\" name=\"address\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>\n";
        echo "<input type=\"submit\" value=\"下一步\">\n";
        echo "<input type=\"reset\" value=\"重新填\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
    }
    else
    {
        $user=$_POST[user];
        $pass=md5($_POST[pass]);
        $email=$_POST[email];
        $msn=$_POST[msn];
        $qq=$_POST[qq];
        $photo=$_POST[photo].".bmp";
        $address=$_POST[address];
        $time=date("Y 年 m 月 d 日");
        require "21-1.php";
```

```
        $sql="select id from $table_user where name='$user'";
        $result=mysql_query($sql,$link) or die(mysql_error());
        $nums=mysql_num_rows($result);
        if($nums!=0)
        {
                echo "<center>";
                echo "<h2>注册的用户名".$user."已经存在！</h2>";
                echo "<h3>请点<a href=# onclick=history.go(-1)>这里</a>返回，重新输入新的用户名！</h3>";
                echo "</center>";
                exit();
        }
        else
        {
                $sql="insert                                                        into
$table_user(name,password,email,photo,msn,qq,address,reg_date)values('$user','$pass','$email','$photo','$ms
n','$qq','$address','$time')";
                mysql_query($sql,$link) or die(mysql_error());
                echo "<center>";
                echo "<h2>新用户".$user."注册成功！</h2>";
                echo "<h3>点<a href=21-5.php>这里</a>进行登录！</h3>";
                echo "</center>";
        }
}
?>
```

把以上代码保存为“21-4.php”。其执行过程相对简单，这里也不再演示。可以参看以前章节中其他类似的内容。

21.4.2　用户登录

用户注册后，如果不登录系统，还是和匿名用户没有区别。只有登录了系统，才能拥有匿名用户所没有的权限，比如购买图书等。用户登录的道理也很简单，判断用户名与密码，如果与表中存在记录相同，则把用户名和 ID 写入 Cookie；反之，给出出错提示并要求重输。在用户成功登录后，再跳转到指定页面即可。

具体内容请参看以下代码：

```
<?php
if($_COOKIE[user])
{
        echo "<html>\n";
        echo "<head>\n";
        echo "<title>注册用户已经登录</title>\n";
        echo "</head>\n";
        echo "<body>\n";
        require "21-3.php";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<tr>\n";
```

```
        echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">注册用户管理</font></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td><a href=\"21-11.php\">修改用户信息</a></td>\n";
        echo "<td><a href=\"21-12.php\">修改用户密码</a></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td><a href=\"21-19.php\">查看我的购物车</a></td>\n";
        echo "<td><a href=\"21-20.php\">查看我的历史订单</a></td>\n";
        echo "</tr>\n";
        echo "</table>";
        echo "<br>";
        require "21-1.php";
        $sql="select admin from $table_user where id='$_COOKIE[id]'";
        $result=mysql_query($sql,$link);
        $rows=mysql_fetch_array($result);
        if($rows[0]==3)
        {
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<tr>\n";
            echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">商品管理</font></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td><a href=\"21-13.php\">增加新的类别</a></td>\n";
            echo "<td><a href=\"21-14.php\">修改已有分类</a></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td><a href=\"21-15.php\">增加新的商品</a></td>\n";
            echo "<td><a href=\"21-16.php\">修改已有商品</a></td>\n";
            echo "</tr>\n";
            echo "<tr>\n";
            echo "<td><a href=\"21-17.php\">查看订单</a></td>\n";
            echo "<td><a href=\"21-18.php\">销售记录</a></td>\n";
            echo "</tr>\n";
            echo "</table>\n";
        }
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
        exit();
}
if(!$_POST[user])                              //如果没有默认参数，则显示 HTML
{
        echo "<html>\n";
        echo "<head>\n";
        echo "<title>注册用户登录</title>\n";
        echo "</head>\n";
        echo "<body>\n";
        require "21-3.php";
        echo "<script language=\"javascript\">\n";
```

```
echo "function juge(theForm)\n";
echo "{\n";
echo "\tif (theForm.user.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入用户名！\");\n";
echo "\t\ttheForm.user.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "\tif (theForm.pass.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入用户密码！\");\n";
echo "\t\ttheForm.pass.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "</script>\n";
echo "<center>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
echo "<tr>\n";
echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">注册用户登录</font></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户名：</td>\n";
echo "<td><input type=\"text\" name=\"user\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>用户密码：</td>\n";
echo "<td><input type=\"password\" name=\"pass\" size=\"21\"></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>选择 COOKIE 有效期：</td>\n";
echo "<td>";
echo "<select name=\"cook_t\" size=\"1\">\n";
echo "<option value=\"1\">最短时效</option>";
echo "<option value=\"2\">1 天</option>";
echo "<option value=\"3\">1 月</option>";
echo "<option value=\"4\">1 年</option>";
echo "</select>";
echo"</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td colspan=\"2\"><center>\n";
echo "<input type=\"submit\" value=\"下一步\">\n";
echo "<input type=\"reset\" value=\"重新填\">\n";
echo "</center></td>\n";
echo "</tr>\n";
echo "</form>\n";
echo "</table>\n";
echo "</center>\n";
```

```
        echo "</body>\n";
        echo "<html>\n";
}
else
{
        $user=$_POST[user];
        $pass=md5($_POST[pass]);
        $cook_t=$_POST[cook_t];
        require "21-1.php";
        $sql="select id from $table_user where name='$user' and password='$pass'";
        $result=mysql_query($sql,$link) or die(mysql_error());
        $nums=mysql_num_rows($result);
        if($nums==0)
        {
                echo "<center>";
                echo "<h2>输入的用户名或者密码错误！ </h2>";
                echo "<h3>请点<a href=# onclick=history.go(-1)>这里</a>返回重新输入！ </h3>";
                echo "</center>";
                exit();

        }
        else
        {
                $rows=mysql_fetch_array($result);
                $id=$rows[id];
                if($cook_t==1)                    //根据不同的 COOKIE 设定时间保存 COOKIE
                {
                        setcookie("user","$user");
                        setcookie("id","$id");
                }
                elseif($cook_t==2)
                {
                        setcookie("user","$user",time()+60*60*24);
                        setcookie("id","$id",time()+60*60*24);
                }
                elseif($cook_t==3)
                {
                        setcookie("user","$user",time()+60*60*24*30);
                        setcookie("id","$id",time()+60*60*24*30);
                }
                else
                {
                        setcookie("user","$user",time()+60*60*24*30*360);
                        setcookie("id","$id",time()+60*60*24*30*360);
                }
                echo "<html>\n";
                echo "<head>\n";
                echo "<title>注册用户登录</title>\n";
                echo "</head>\n";
                echo "<body>\n";
```

```
                require "21-3.php";
                echo "<h2>用户".$user."登录成功！ </h2>";
                echo "<h3>两秒后进入商城首页面！ </h3>";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=21-6.php\">";
                echo "</center>";
        }
}
?>
```

将上述代码保存为“21-5.php”，以备用户登录之用。由于用户登录过程也比较简单，这里也不再给出演示。如果要浏览相似的执行结果，请查看本书第 20 章相关内容即可。

21.5　前台显示界面

这一节来完成所用的前台界面，即普通用户能够看到的界面。其中包括首页页面、图书列表页面、按种类查看页面、某一图书详细情况页面、购物车的实现等。

下面就来一个一个地实现这些所需功能。

21.5.1　首页面的实现

首页面是用户进入系统后看到的第 1 个页面，是进入其他页面的前提。这一节来完成系统首页面的设计。该首页面包括以下内容：用户登录的接口、总图书数、注册用户人数、图书的分类查看、最新添加的图书、图书的搜索界面等。由于包括内容比较多，所以说首页面也是一个相对复杂的页面。

下面给出具体的代码来实现这些内容的显示。代码如下：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>网上图书商城首页</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
echo "<table width=80%>\n";
echo "<tr>\n";
echo "<td width=\"20%\">\n";
if(!$_COOKIE[user])                                    //如果没用用户登录，则显示 HTML
{
        echo "<script language=\"javascript\">\n";
        echo "function juge(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.user.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入用户名！ \");\n";
        echo "\t\ttheForm.user.focus();\n";
```

```
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.pass.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入用户密码！\");\n";
        echo "\t\ttheForm.pass.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "function juge2(theForm)\n";
        echo "{\n";
        echo "\tif (theForm.search_c.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入搜索内容！\");\n";
        echo "\t\ttheForm.search_c.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "</script>\n";
        echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\">\n";
        echo "<form method=\"post\" action=21-5.php onsubmit=\"return juge(this)\">\n";
        echo "<tr>";
        echo "<td><center>用户登录</center></td>\n";
        echo "</tr>\n";
        echo "<tr>";
        echo "<td>用户名：";
        echo "<input type=\"text\" name=\"user\" size=\"6\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>密  码：";
        echo "<input type=\"password\" name=\"pass\" size=\"5\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td><center>\n";
        echo "<td><center><a href=\"21-4.php\">注册</a>\n<input type=\"submit\" value=\"登录\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>";
        echo "</table>";
}
else                                                    //如果有用户登录，则显示操作连接
{
        echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\">\n";
        echo "<tr>";
        echo "<td>登录用户：".$_COOKIE[user]."</td>\n";
        echo "</tr>\n";
        echo "<tr>";
        echo "<td><center><a href=\"21-15.php\">退出登录</a></center></td>\n";
        echo "</tr>\n";
        echo "</table>";
```

```
}
echo "<br>";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\">\n";
echo "<tr>";
echo "<td><center>本站信息</center></td>\n";
echo "</tr>\n";
echo "<tr>";
echo "<td>共有用户：";
require "21-1.php";
$sql="select id from $table_user";
$result=mysql_query($sql,$link);
$num=mysql_num_rows($result);
echo $num;
echo "名</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>共有图书：";
$sql="select id from $table_book";
$result=mysql_query($sql,$link);
$num=mysql_num_rows($result);
echo $num;
echo "种</td>\n";
echo "</tr>\n";
echo "</table>";
echo "<br>";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\">\n";
echo "<tr>";
echo "<td>最新用户：";
$sql="select name from $table_user order by id desc";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
echo $rows[0];
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>最新图书：";
$sql="select * from $table_book order by id desc";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
echo "<a href=21-10.php?id=".$rows[id].">".$rows[book_name]."</a>";
echo "</td>\n";
echo "</tr>\n";
echo "</table>";
echo "<br>";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\">\n";
echo "<tr>";
echo "<td>图书分类:</td>\n";
echo "</tr>\n";
$sql2="select * from $table_type where p_id!=0";
$result2=mysql_query($sql2,$link);
```

```
while($rows2=mysql_fetch_array($result2))
{
	echo "<tr>\n";
	echo "<td>";
	echo "<a href=21-8.php?id=".$rows2[id].">".$rows2[type_name]."</a>：（".$rows2[type_num]."）";
	echo "</td>\n";
	echo "</tr>\n";
}
echo "</table>";
echo "</td>\n";
echo "<td width=\"80%\">\n";
echo "<script language=\"javascript\">";
echo "function juge2(theForm)\n";
echo "{\n";
echo "\tif (theForm.search_c.value == \"\")\n";
echo "\t{\n";
echo "\t\talert(\"请输入搜索内容！\");\n";
echo "\t\ttheForm.search_c.focus();\n";
echo "\t\treturn (false);\n";
echo "\t}\n";
echo "}\n";
echo "</script>\n";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\" height=\"100%\">\n";
echo "<tr>";
echo "<form method=\"post\" action=\"21-9.php\" onsubmit=\"return juge2(this)\">\n";
echo "<td>搜索图书</td>\n";
echo "<td>搜索内容：";
echo "<input type=\"text\" name=\"search_c\" size=\"6\">";
echo "搜索类型：";
echo "<select name=\"search_t\" size=\"1\">\n";
echo "<option value=\"book_name\">书名</option>";
echo "<option value=\"book_author\">作者</option>";
echo "<option value=\"book_pub\">出版社</option>";
echo "</select>";
echo "<input type=\"submit\" value=\"搜索\">\n";
echo "</td>\n";
echo "</form>\n";
echo "</tr>\n";
echo "</table>";
echo "<br>";
echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"100%\">\n";
if($num!=0)
{echo "<tr>";
echo "<td colspan=\"2\">最新推荐图书</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td width=\"30%\">图书名称：</td>\n";
echo "<td>".$rows[book_name]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
```

```
echo "<td>图书作者：</td>\n";
echo "<td>".$rows[book_author]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>出版社：</td>\n";
echo "<td>".$rows[book_pub]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>售价：</td>\n";
echo "<td>".$rows[book_cost]."元</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>所属类别：</td>\n";
echo "<td>".$rows[book_type]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>该书的数量：</td>\n";
echo "<td>".$rows[book_num]."本</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>该书的简介：</td>\n";
echo "<td>".$rows[book_description]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>该书的封面扫描图：</td>\n";
echo "<td>";
if(!$rows[book_photo])
{
	$rows[book_photo]="images/nopic.gif";
}
echo "<img src=\"".$rows[book_photo]."\">";
echo "</td>\n";
echo "</tr>\n";
echo "</table>";
echo "</td>\n";
echo "</tr>\n";
}
else
{
	echo "<tr>\n";
	echo "<td>暂时没有推荐图书</td>\n";
	echo "</tr>\n";
}
echo "</table>\n";
echo "</center>\n";
echo "</body>\n";
echo "</html>\n";
?>
```

把上述代码保存为“21-6.php”，首页面创建完毕。首先在首页用管理员身份进行登录。首页将会出现如图 21.3 所示的结果，也说明该程序正常运行。



图 21.3　管理员登录后的首页执行结果

21.5.2　图书列表页面的实现

图书列表页面，按照添加日期的先后，显示出所有图书表中的记录，如果记录数超过了每页显示数，则分页显示。显示信息有：书名、作者、简介（如果简介太长，则截取其部分显示）。

下面是具体代码内容：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>查看所有图书</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
require "21-1.php";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
$sql="select id from $table_book";                    //从列表中读出所有图书记录
$result=mysql_query($sql,$link) or die(mysql_error());              //发送查找列表请求
$num=mysql_num_rows($result);                         //获取结果条数
$p_count=ceil($num/10);                               //总页数
if ($_GET["page"]==0 && !$_GET["page"]) $page=1;//当前页
else $page=$_GET["page"];
if($num<1)                                            //如果没有记录
{
    echo "<tr>\n";
    echo "<td>";
    echo "<center><h2>暂时还没有图书的记录</h2></center>";          //输出相应信息
    echo "</td>\n";
    echo "</tr>\n";
    exit();                                           //退出所有 PHP 代码
}
else                                                  //如果有记录，则执行相应操作
{
    $s=($page-1)*10;
    $sql="select * from $table_book order by id limit $s,10";
    $result=mysql_query($sql,$link);
    echo "<tr>\n";
    echo "<td>书名</td>\n";
    echo "<td>作者</td>\n";
    echo "<td>价格</td>\n";
    echo "<td>类别</td>\n";
```

```
        echo "<td>简介</td>\n";
        while($rows=mysql_fetch_array($result))        //循环显示记录内容
        {
            echo "<tr>\n";
            echo "<td><a href=\"21-10.php?id=".$rows[id]."\">".$rows[book_name]."</a></td>\n";    //显示书名
            echo "<td>".$rows[book_author]."</td>\n";//显示作者
            echo "<td>".$rows[book_cost]."</td>\n";  //显示价格
            $sql2="select id,type_name from $table_type where id='$rows[book_type]'";
            $result2=mysql_query($sql2,$link);
            $rows2=mysql_fetch_array($result2);
            echo "<td><a href=\"21-8.php?id=".$rows2[0]."\">".$rows2[1]."</a></td>";        //显示类别
            if(strlen($rows[book_description])>100)
            $rows[book_description]=substr($rows[description],0,100);
            echo "<td>".$rows[book_description]."</td>\n";
            echo "</tr>\n";
        }
    }
    echo "</table>";
                                //以下为分页显示内容
    $prev_page=$page-1;
    $next_page=$page+1;
    echo "                  <p align=\"center\"> ";
    if ($page>1)
    {
        echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
    }
    if ($prev_page>=1)
    {
        echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
    }
    if ($next_page<=$p_count)
    {
        echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
    }
    if ($page<$p_count)
    {
        echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
    }
    echo "</center>";
    echo "</body>";
    echo "</html>";
    ?>
```

把以上代码保存为“21-7.php”，以便用户查看所有记录时调用该文件。该 PHP 执行机理相当简单，不再给出效果图。

21.5.3　按种类查看页面

该页面显示图书的所有种类，并且在每个子种类上都有指向该种类查看页面的超链接，用户只需

要单击该超链接，就可以实现查看该种类下的所有图书。

这里把所有种类的显示与某一种类的单独显示功能整合到了一个页面。用一个参数来判断要显示哪一类。如果没有任何参数，则显示所有种类；如果指定了参数 ID，则显示某一类的结果。

下面给出具体代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>按种类查看图书</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
if(!$_GET[id])
{
	echo "<font size=5>查看所有种类</font>";
	echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
	require "21-1.php";
	$sql="select * from $table_type where p_id=0";
	$result=mysql_query($sql,$link);
	while($rows=mysql_fetch_array($result))			//循环显示主类别
	{
		echo "<tr>\n";
		echo "<td colspan=\"2\">";
		echo $rows[type_name];
		echo "（".$rows[type_num]."）";
		echo "</td>\n";
		echo "</tr>\n";
		$i=0;
		$sql2="select * from $table_type where p_id='$rows[id]' and id>'$rows[id]'";
		$result2=mysql_query($sql2,$link) or die(mysql_error());
		$m_count=mysql_num_rows($result2);
		while($rows2=mysql_fetch_array($result2))	//循环显示主类别下的分类别
		{
			if($i%2==0) echo "<tr>\n";
			echo "<td width=\"50%\">";
			echo "<a href=21-8.php?id=".$rows2[id].">".$rows2[type_name]."</a>";
			echo "（".$rows2[type_num]."）";
			echo "</td>\n";
			$i++;
			if(($m_count%2==1) and $i==($m_count))
			echo "<td> </td>";
			if($i%2==1) echo "</tr>\n";
		}
	}
}
else
{
	require "21-1.php";
```

```
    $sql="select type_name from $table_type where id='$_GET[id]'";
    $result=mysql_query($sql,$link);
    $type_name=mysql_fetch_array($result);
    echo "<font size=5>查看种类：".$type_name[0]."</font>";
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    $sql="select * from $table_book where book_type='$_GET[id]' order by id desc";    //从列表中读出所有图
书记录
    $result=mysql_query($sql,$link) or die(mysql_error());                  //发送查找列表请求
    $num=mysql_num_rows($result);                           //获取结果条数
    $p_count=ceil($num/10);                                 //总页数
    if ($_GET["page"]==0 && !$_GET["page"]) $page=1;//当前页
    else $page=$_GET["page"];
    if($num<1)                                              //如果没有记录
    {
        echo "<tr>\n";
        echo "<td>";
        echo "<center><h2>暂时还没有图书的记录</h2></center>";         //输出相应信息
        echo "</td>\n";
        echo "</tr>\n";
        exit();                                             //退出所有 PHP 代码
    }
    else                                                    //如果有记录，则执行相应操作
    {
        echo "<tr>\n";
        echo "<td>书名</td>\n";
        echo "<td>作者</td>\n";
        echo "<td>价格</td>\n";
        echo "<td>类别</td>\n";
        echo "<td>简介</td>\n";
        while($rows=mysql_fetch_array($result))             //循环显示记录内容
        {
            echo "<tr>\n";
            echo "<td><a href=\"21-10.php?id=".$rows[id]."\">".$rows[book_name]."</a></td>\n";     //显示
书名
            echo "<td>".$rows[book_author]."</td>\n";//显示作者
            echo "<td>".$rows[book_cost]."</td>\n";    //显示价格
            $sql2="select type_name from $table_type where id='$rows[book_type]'";
            $result2=mysql_query($sql2);
            $rows2=mysql_fetch_array($result2);
            echo "<td>".$rows2[0]."</td>";                  //显示类别
            if(strlen($rows[book_description])>100)
            $rows[book_description]=substr($rows[description],0,100);
            echo "<td>".$rows[book_description]."</td>\n";
            echo "</tr>\n";
        }
    }
    echo "</table>";
                                                            //以下为分页显示内容
    $prev_page=$page-1;
    $next_page=$page+1;
```

```
        echo "                    <p align=\"center\"> ";
        if ($page>1)
        {
            echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
        }
        if ($prev_page>=1)
        {
            echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
        }
        if ($next_page<=$p_count)
        {
            echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
        }
        if ($page<$p_count)
        {
            echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
        }
        echo "</center>";
        echo "</body>";
        echo "</html>";
    }
    ?>
```

把此代码保存为“21-8.php”，以方便用户查看类别时调用该文件。在 PHP 环境中执行该文件（前提是以管理员身份登录系统并添加了相应类别及图书），其执行结果如图 21.4 所示。

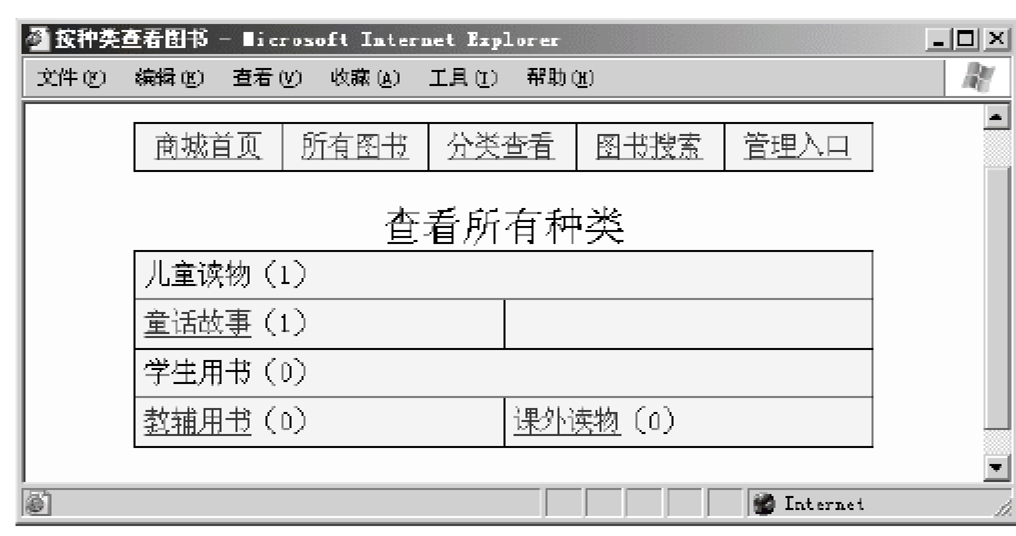


图 21.4　查看所有分类执行结果

21.5.4　搜索图书页面

如果图书的种类太多，给用户提供一个搜索的平台是有必要的。用户可以按照图书的名称、作者、出版社、及所属类型等各项信息来查找相应的目标图书。如果找到了相应的图书，就循环显示所有查找结果，并给出相应的查看该书详情的链接。

下面是实现搜索功能的具体代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>图书搜索</title>\n";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
if(!$_POST[search_c])                          //如果没有默认参数，则显示 HTML
{
    echo "<script language=\"javascript\">\n";
    echo "function juge(theForm)\n";
    echo "{\n";
```

```
        echo "\tif (theForm.search_c.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入搜索内容！ \");\n";
        echo "\t\ttheForm.search_c.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "</script>\n";
        echo "<center>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\" align=\"center\"><font size=\"5px\">图书搜索</font></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入搜索内容：</td>\n";
        echo "<td><input type=\"text\" name=\"search_c\"></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择搜索类型：</td>\n";
        echo "<td>";
        echo "<select name=\"search_t\" size=\"1\">\n";
        echo "<option value=\"book_name\">书名</option>";
        echo "<option value=\"book_author\">作者</option>";
        echo "<option value=\"book_pub\">出版社</option>";
        echo "</select>";
        echo"</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择搜索模式：</td>\n";
        echo "<td>";
        echo "<input type=\"radio\" name=\"search_m\" value=\"1\" checked>精确查找";
        echo "<input type=\"radio\" name=\"search_m\" value=\"2\">模糊查找";
        echo"</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>\n";
        echo "<input type=\"submit\" value=\"下一步\">\n";
        echo "<input type=\"reset\" value=\"重新填\">\n";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "<html>\n";
    }
    else
    {
        require "21-1.php";                              //调用配置文件
```

```
$search_c=$_POST[search_c];                              //获取表单变量
$search_t=$_POST[search_t];
$search_m=$_POST[search_m];
if($search_m=="1")                                       //根据搜索模式的不同设置不同的搜索类型
{
    $sql="select * from $table_book where $search_t='$search_c'";
}
else
{
    $sql="select * from $table_book where $search_t like '$search_c'";
}
$result=mysql_query($sql,$link);                         //发送 SQL 请求
$num=mysql_num_rows($result);                            //获取结果数
if($num<1)                                               //如果没有结果，则显示内容
{
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    echo "<tr>\n";
    echo "<td align=\"center\"><font size=\"5px\">图书搜索</font></td>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td  align=\"center\">对不起没有找到你所要求的内容！</td>\n";
    echo "</tr>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td  align=\"center\">点<a href=# onclick=history.go(-1)>这里</a>返回</td>\n";
    echo "</tr>\n";
    echo "</table>";
}
else                                                     //如果有结果，则循环显示内容
{
    echo "共找到".$num."条记录";
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    echo "<tr>\n";
    echo "<td>书名</td>\n";
    echo "<td>作者</td>\n";
    echo "<td>价格</td>\n";
    echo "<td>类别</td>\n";
    echo "<td>简介</td>\n";
    while($rows=mysql_fetch_array($result))
    {
        echo "<tr>\n";
        echo "<td><a href=\"21-10.php?id=".$rows[id]."\">".$rows[book_name]."</a></td>\n"; //显示书名
        echo "<td>".$rows[book_author]."</td>\n";//显示作者
        echo "<td>".$rows[book_cost]."</td>\n";   //显示价格
        $sql2="select type_name from $table_type where id='$rows[book_type]'";
        $result2=mysql_query($sql2);
        $rows2=mysql_fetch_array($result2);
        echo "<td>".$rows2[0]."</td>";           //显示类别
        if(strlen($rows[book_description])>100)
        $rows[book_description]=substr($rows[description],0,100);
```

```
                    echo "<td>".$rows[book_description]."</td>\n";
                    echo "</tr>\n";
                }
                echo "</table>";
            }
            echo "</center>";
            echo "</body>";
            echo "</html>";
        }
        ?>
```

将以上代码保存为“21-9.php”。这样，用户需要对图书资源进行搜索时，只需要单击相应的超链接，就可以进入该页面以实现对图书的搜索了。

21.5.5　查看图书详情页面

不管是在图书列表页、种类查看页，还是图书搜索结果页面，当用户看到了中意的图书后，都可以单击该图书上的超链接，以查看该书的详细情况。查看图书详情页包含该书的所有详细情况。用户可以选择购买该图书从而启动购物车功能。

这里来说说购物车的实现原理。购物车可以采用多种实现机制，如 Cookie、Session 或者隐藏帧等。这里将采用 Cookie 方式。当用户单击“把该书放入购物车”链接后，程序将在后台把该书记录写入 Cookie。这里的 Cookie 写入工作采用 JavaScript 来实现，因为使用 JavaScript 可以减少服务器的负担。有关 JavaScript 操作 Cookie 的函数，请读者去自行查找相应内容。

下面先给出要调用的 JS 文件的具体代码：

```
function SetCookie (name, value)     //设置名称为 name,值为 value 的 Cookie
{var expdate = new Date();
expdate.setTime(expdate.getTime() + 30 * 60 * 1000);
document.cookie = name+"="+value+";expires="+expdate.toGMTString()+";path=/";
alert("添加商品"+name+"成功!");
var
cat=window.open("21-19.php","cat","toolbar=no,menubar=no,location=no,status=no,width=420,height=280");  //
打开一个新窗口来显示统计的商品信息，即显示“手推车”}
}
function Deletecookie (name) {        //删除名称为 name 的 Cookie
var exp = new Date();
    exp.setTime (exp.getTime() - 1);
    var cval = GetCookie (name);
    document.cookie = name + "=" + cval + "; expires=" + exp.toGMTString();
}
function Clearcookie()                //清除 COOKIE
    {
    var temp=document.cookie.split(";");
    var loop3;
    var ts;
    for (loop3=0;loop3<temp.length;loop3++)
```

```
            {
            ts=temp[loop3].split("=")[0];
            if (ts.indexOf('mycat')!=-1)
                DeleteCookie(ts);        //如果 ts 含“mycat”则执行清除
            }
        }

function getCookieVal (offset) {        //取得项名称为 offset 的 cookie 值
        var endstr = document.cookie.indexOf (";", offset);
        if (endstr == -1)
            endstr = document.cookie.length;
            return unescape(document.cookie.substring(offset, endstr));
}

function GetCookie (name) {             //取得名称为 name 的 cookie 值
            var arg = name + "=";
            var alen = arg.length;
            var clen = document.cookie.length;
            var i = 0;
            while (i < clen) {
            var j = i + alen;
            if (document.cookie.substring(i, j) == arg)
                    return getCookieVal (j);
                    i = document.cookie.indexOf(" ", i) + 1;
                    if (i == 0) break;
            }
            return null;
}
```

先将以上代码保存为 mycat.js，以方便程序调用该 JS 文件。
下面给出查看图书详情的代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>查看图书详情图书</title>\n";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
if(!$_GET[id])                                    //如果没有用户请求，则显示信息
{
      echo "没有请求 ID！ <br>";
      echo "点<a href=\"21-6.php\">这里</a>返回首页！ ";
}
else                                              //如果有用户请求，则执行操作
{
      echo "<script language=\"javascript\" src=\"mycat.js\">\n";
      echo "</script>";
      require "21-1.php";
```

```
$sql="select * from $table_book where id='$_GET[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
echo "<tr>\n";
echo "<td colspan=\"2\"><center><h2>查看图书详情</h2></center></td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td width=\"30%\">图书名称：</td>\n";
echo "<td>".$rows[book_name]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>图书作者：</td>\n";
echo "<td>".$rows[book_author]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>出版社：</td>\n";
echo "<td>".$rows[book_pub]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>售价：</td>\n";
echo "<td>".$rows[book_cost]."元</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>所属类别：</td>\n";
echo "<td>".$rows[book_type]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>该书的数量：</td>\n";
echo "<td>".$rows[book_num]."本</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>该书的简介：</td>\n";
echo "<td>".$rows[book_description]."</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td>该书的封面扫描图：</td>\n";
echo "<td>";
if(!$rows[book_photo])
{
	$rows[book_photo]="images/nopic.gif";
}
echo "<img src=\"".$rows[book_photo]."\">";
echo "</td>\n";
echo "</tr>\n";
echo "<tr>\n";
echo "<td colspan=\"2\" align=\"center\"><input type=\"button\" value=\" 把 该 书 加 入 购 物 车 \" 
onclick=SetCookie(\"cat".$rows[id]."\",\"1\")></td>\n";
echo "</tr>\n";
echo "</table>\n";
```

```
        echo "</center>\n";
        echo "</body>";
        echo "</html>\n";
    }
    ?>
```

将以上代码保存为“21-10.php”，在 PHP 执行环境下运行该文件（后面必须带有相应图书的 ID 号），其执行结果如图 21.5 所示。

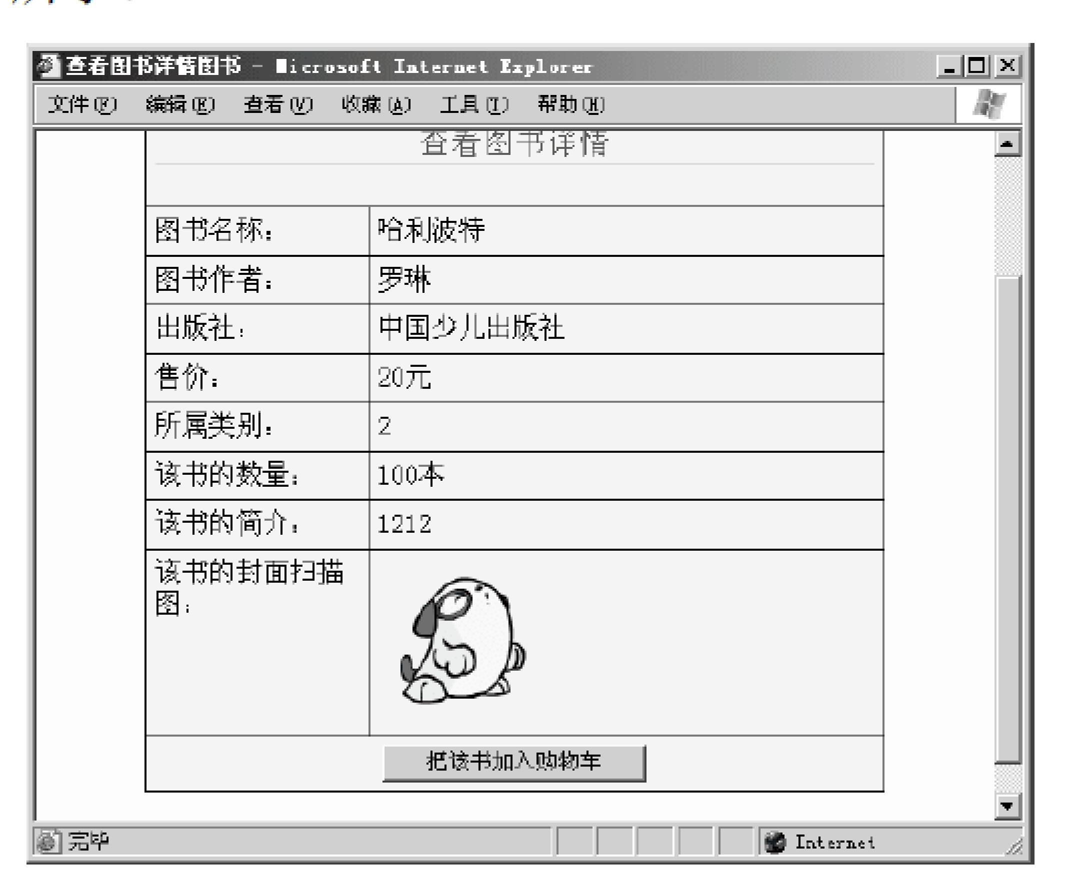


图 21.5　查看图书详情页面执行结果

从图 21.5 可以发现，显示出了图书的所有详细信息，并且用户还可以通过单击“把该书放入购物车”按钮来启动购物车。

21.6　购物车的实现

在 21.5 节的查看图书详情页面中已经出现了购物车。该程序中的购物车采用 Cookie 这样的实现机制。

这一节来详细介绍，如何实现购物车，及让用户的提交内容转化为实实在在的订单。

21.6.1　查看当前购物车

该页面要实现的功能是统计系统中的当前站点的 Cookie 值。如果是描述商品的 Cookie，则记录下该 Cookie 值，并读取库表中相应商品的信息显示给用户。用户可以自由选择是否购买已经加入购物车的商品，并且可以随便输入购买的数量。

下面给出查看当前购物车的详细代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>查看购物车</title>\n";
```

```
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
echo "<LINK href=\"style.css\" rel=stylesheet>\n";
if(!$_POST[mycat])                                    //如果没有用户提交显示内容
{
        require "21-1.php";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"$PATH_INFO\">\n";
        echo "<input type=\"hidden\" name=\"mycat\" value=\"post\">";
        echo "<tr>\n";
        echo "<td colspan=\"4\"><center><h2>您的购物车信息</h2></center></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择</td>\n";
        echo "<td>名称</td>\n";
        echo "<td>单价</td>\n";
        echo "<td>数量</td>\n";
        echo "</tr>\n";
        $temp=array_keys($_COOKIE);
        $j=0;
        for($i=0;$i<count($temp);$i++)
        {
                if(ereg("cat",$temp[$i]))                    //查找已添加到购物车的商品
                {
                        $j++;
                        $catid=ereg_replace("cat","",$temp[$i]);
                        $sql="select * from $table_book where id='$catid'";
                        $result=mysql_query($sql,$link);
                        $rows=mysql_fetch_array($result);
                        echo "<input type=\"hidden\" name=\"id[]\" value=\"".$rows[id]."\">\n";
                        echo "<tr>\n";
                        echo "<td><input type=\"checkbox\" name=\"c".$j."\"></td>\n";
                        echo "<td>".$rows[book_name]."</td>\n";
                        echo   "<td><input   type=\"text\"   value=\"".$rows[book_cost]."\"   name=\"m[]\"   readonly
size=\"5\"></td>\n";
                        echo "<td><input type=\"text\" name= \"t[]\" value=\"1\" size=\"3\"></td>\n";
                        echo "</tr>\n";
                }
        }
        echo "<tr>\n";
        echo "<td colspan=\"4\"><center>";
        echo "<input type=\"submit\" value=\"结帐\">";
        echo "<input type=\"button\" value=\"继续购物\" onclick=window.close()>";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>";
        echo "</table>";
}
else
```

```
{
	$id=$_POST[id];
	$m=$_POST[m];
	$t=$_POST[t];
	$time=date("Y 年 m 月 d 日");
	require "21-1.php";
	echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
	echo "<tr><td colspan=\"4\"><center>您选购了以下商品:</center></td></tr>";
	echo "<tr>";
	echo "<td>书名</td>";
	echo "<td>单价</td>";
	echo "<td>数量</td>";
	echo "<td>小计</td>";
	echo "</tr>";
	$j=0;
	for($i=1;$i<=count($id);$i++)                    //循环显示所有商品
	{
		$c="c".$i;
		if($$c!="")
		{
			$temp=$id[$i-1];
			$temp2=$m[$i-1];
			$temp3=$t[$i-1];
			$sql="select * from $table_book where id='$temp'";
			$result=mysql_query($sql,$link);
			$rows=mysql_fetch_array($result);
			echo "<tr>";
			echo "<td>".$rows[book_name]."</td>";
			echo "<td>".$temp2."</td>";
			echo "<td>".$temp3."</td>";
			$z[$j]=$m[$i-1]*$t[$i-1];
			$temp4=$z[$j];
			echo "<td>".$z[$j]."</td>";
			echo "</tr>";
			$j++;
			$sql="insert                                                    into
$table_order(order_user_id,order_book_id,order_book_num,order_user_name,order_cost,order_date)
values('$_COOKIE[id]','$temp','$temp3','$_COOKIE[user]','$temp4','$time')";
			mysql_query($sql,$link);
		}
	}
	for($i=0;$i<count($z);$i++)
	{
		$s=$s+$z[$i];
	}
	echo "<tr><td colspan=\"4\"><center>总计:".$s."</center></td></tr>";
	echo  "<tr><td  colspan=\"4\"> 已 经 生 成 订 单 , 点 <input  type=\"button\"  value=\" 这 里 结 束 操 作 \"
onclick=window.close></td></tr>";
}
?>
```

将以上代码保存为“21-19.php”。该文件具备了查看购物车、选择选购的商品、提交购物车信息等功能。它是购物车功能的主体文件。如果用户在浏览商品时，把某一商品加入购物车，系统会自动以弹出窗口的形式执行 21-19.php，以显示用户当前选定的商品。其执行结果如图 21.6 所示。

其中显示出了用户所选择的商品及其价格。用户可以选中需要购买的商品，单击“结账”按钮提交购物车，也可以单击“继续购物”按钮，关闭该窗口。



图 21.6　查看购物车页面执行结果

21.6.2　查看用户历史订单

用户在提交购物车后，购物信息将生成订单，所以应该使用户有权限能查看所有的历史订单。该功能实现也很简单，从订单表里选取用户名为当前登录用户的记录显示出来即可。

下面给出具体实现代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>查看登录用户历史订单</title>\n";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
if(!$_COOKIE[user])
{
	echo "你没有登录,没有权限执行这项操作！<p>";
	echo "点<a href=\"21-5.php\">这里</a>进行登录";
	exit();
}
else
{
	require "21-1.php";
	echo "<h2>查看用户".$_COOKIE[user]."的订单记录</h2>";
	$sql="select id from $table_order where order_user_id='$_COOKIE[id]'";	//从列表中读出所有图书记录
	$result=mysql_query($sql,$link) or die(mysql_error());		//发送查找列表请求
	$num=mysql_num_rows($result);			//获取结果条数
	$p_count=ceil($num/10);				//总页数
	if ($_GET["page"]==0 && !$_GET["page"]) $page=1;//当前页
	else $page=$_GET["page"];
	echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
	if($num<1)					//如果没有记录
	{
		echo "<tr>\n";
		echo "<td>";
		echo "<center><h2>暂时还没有该用户的订单记录</h2></center>";	//输出相应信息
		echo "</td>\n";
```

```
            echo "</tr>\n";
            exit();                                     //退出所有 PHP 代码
        }
        else                                            //如果有记录，则执行相应操作
        {
            echo "<tr>\n";
            echo "<td>购书 ID</td>\n";
            echo "<td>购书数量</td>\n";
            echo "<td>购书总额</td>\n";
            echo "<td>订单状态</td>\n";
            echo "<td>提交日期</td>\n";
            echo "</tr>\n";
            $s=($page-1)*10;
            $sql="select * from $table_order where order_user_id='$_COOKIE[id]' order by id limit $s,10";
            $result=mysql_query($sql,$link);
            while($rows=mysql_fetch_array($result))
            {
                echo "<tr>\n";
                echo "<td>".$rows[order_book_id]."</td>\n";
                echo "<td>".$rows[order_book_num]."</td>\n";
                echo "<td>".$rows[order_cost]."</td>\n";
                echo "<td>".$rows[order_state]."</td>\n";
                echo "<td>".$rows[order_date]."</td>\n";
                echo "</tr>\n";
            }
            echo "</table>";
            $prev_page=$page-1;
            $next_page=$page+1;
            echo "              <p align=\"center\"> ";
            if ($page>1)
            {
                echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
            }
            if ($prev_page>=1)
            {
                echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
            }
            if ($next_page<=$p_count)
            {
                echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
            }
            if ($page<$p_count)
            {
                echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
            }
            echo "</center>";
            echo "</body>";
            echo "</html>";
        }
    }
?>
```

把以上代码保存为“21-20.php”。这里可以方便用户随时查看当前用户的历史订单记录。

21.7　管理功能的实现

这里的管理功能包括两层含义：一是普通用户可以更改自己的注册信息、密码及查看购物车等内容，另一层含义是管理员可以添加或者修改图书类别、添加修改新的图书、查看处理订单、查看销售记录等。

可以看出，不管对于普通用户还是管理者，后台的管理功能都是十分重要的。这一节就来逐个实现这些后台管理功能。这里采用这样的形式，所有管理操作的链接都集中在一个页面上，即用户登录页，用户在登录前显示的是登录界面，而在登录之后，则显示所有操作的链接，如图 21.7 所示。

通过这个界面，普通用户，或者管理员可以执行相应的操作。当然只有管理员才能执行的操作，普通用户是看不到的。



图 21.7　管理员登录后的“管理入口”执行结果

21.7.1　更改用户信息

注册用户的信息在注册后并不是一成不变的，用户可以自由更改自己的信息，而用户信息的更改的实质就是改变表中特定项的记录。先给出用户一个人机交互界面，要求用户输入更改的内容，内容输入后，转到后台，对相应项进行更改操作。

下面给出具体代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改注册用户信息</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
if(!$_COOKIE[user])
{
      echo "您还没有登录！<p>";
      echo "点<a href=\"21-5.php\">这里</a>进行登录";
      exit();
}
else
{
```

```
if(!$_POST[email])
{
	echo "<script language=\"javascript\">\n";
	echo "function juge(theForm)\n";
	echo "{\n";
	echo "\tif (theForm.email.value == \"\")\n";
	echo "\t{\n";
	echo "\t\talert(\"请输入邮箱！\");\n";
	echo "\t\ttheForm.email.focus();\n";
	echo "\t\treturn (false);\n";
	echo "\t}\n";
	echo "}\n";
	echo "function s_photo(the)\n";
	echo "{\n";
	echo "\tdocument.img.src='images/'+the.photo.value+'.bmp';\n";
	echo "}\n";
	echo "</script>\n";
	require "21-1.php";
	$sql="select * from $table_user where id='$_COOKIE[id]'";
	$result=mysql_query($sql,$link);
	$rows=mysql_fetch_array($result);
	echo "<table cellpadding=\"1\" cellspacing=\"1\" width=\"80%\">\n";
	echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
	echo "<tr>";
	echo "<td colspan=\"2\">以下几项是可以修改的：</td>\n";
	echo "</tr>\n";
	echo "<tr>\n";
	echo "<td>用户地址：</td>\n";
	echo "<td><input type=\"text\" name=\"address\" value=\"".$rows[address]."\"></td>\n";
	echo "</tr>\n";
	echo "<tr>\n";
	echo "<td>用户邮箱：</td>\n";
	echo "<td><input type=\"text\" name=\"email\" value=\"".$rows[email]."\"></td>\n";
	echo "</tr>\n";
	echo "<tr>\n";
	echo "<td>用户 QQ：</td>\n";
	echo "<td><input type=\"text\" name=\"qq\" value=\"".$rows[qq]."\"></td>\n";
	echo "</tr>\n";
	echo "<tr>\n";
	echo "<td>用户 MSN：</td>\n";
	echo "<td><input type=\"text\" name=\"msn\" value=\"".$rows[msn]."\"></td>\n";
	echo "</tr>\n";
	echo "<tr>\n";
	echo "<td>用户图像：</td>\n";
	echo "<td>";
	echo "<select name=\"photo\" size=\"1\" onchange=\"s_photo(this.form)\">\n";
	for($i=1;$i<21;$i++)
	{
		echo "<option value=".$i.">".$i."</option>\n";;
```

```
			}
				echo "</select>\n";
			echo "<img src=\"images/".$rows[photo]."\" name=\"img\">\n";
			echo "</td>\n";
			echo "</tr>\n";
			echo "<tr>";
			echo "<td colspan=\"2\"><center><input type=\"submit\" value=\"确认提交\"></center></td>\n";
			echo "</tr>\n";
			echo "</form>\n";
			echo "</table>\n";
		}
		else
		{
			$email=$_POST[email];
			$address=$_POST[address];
			$qq=$_POST[qq];
			$msn=$_POST[msn];
			$photo=$_POST[photo].".bmp";
			require "21-1.php";
			$sql="update                    $table_user                    set
	email='$email',address='$address',qq='$qq',msn='$msn',photo='$photo' where id='$_COOKIE[id]'";
			if(mysql_query($sql,$link))
			{
				echo "修改注册信息成功，现在返回首页";
				echo "<meta http-equiv=\"refresh\" content=\"2; url=21-6.php\">";
			}
			else
			{
				echo "修改注册信息失败，现在返回更改信息页";
				echo "<meta http-equiv=\"refresh\" content=\"2; url=21-11.php\">";
			}
		}
	}
	?>
```

把以上代码保存为“21-11.php”。这样，当用户需要对注册信息进行更改时就可以调用该页面。

21.7.2　更改用户密码

密码始终是一个比较敏感的选项，因为它是用户进入系统的一个钥匙。所以要把密码单独放出来更改。不过更改密码的原理同更改普通信息一样，对用户的输入进行判断，如果符合条件，就用新密码替换旧密码。

具体过程请参考以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改注册用户密码</title>\n";
```

```
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
if(!$_COOKIE[user])
{
	echo "你还没有登录！<p>";
	echo "点<a href=\"21-5.php\">这里</a>进行登录";
	exit();
}
else
{
	if(!$_POST[password])
	{
		echo "<script language=\"javascript\">\n";
		echo "function juge(theForm)\n";
		echo "{\n";
		echo "\tif (theForm.password.value == \"\")\n";
		echo "\t{\n";
		echo "\t\talert(\"请输入旧密码！\");\n";
		echo "\t\ttheForm.password.focus();\n";
		echo "\t\treturn (false);\n";
		echo "\t}\n";
		echo "\tif (theForm.newpassword.value == \"\")\n";
		echo "\t{\n";
		echo "\t\talert(\"请输入新密码！\");\n";
		echo "\t\ttheForm.newpassword.focus();\n";
		echo "\t\treturn (false);\n";
		echo "\t}\n";
		echo "\tif (theForm.newpassword.value.length <8)\n";
		echo "\t{\n";
		echo "\t\talert(\"密码要在 8 位以上！\");\n";
		echo "\t\ttheForm.newpassword.focus();\n";
		echo "\t\treturn (false);\n";
		echo "\t}\n";
		echo "\tif (theForm.newpassword.value != theForm.repassword.value)\n";
		echo "\t{\n";
		echo "\t\talert(\"重复输入的密码不一致！\");\n";
		echo "\t\ttheForm.repassword.focus();\n";
		echo "\t\treturn (false);\n";
		echo "\t}\n";
		echo "}\n";
		echo "</script>\n";
		echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
		echo "<form method=\"post\" action=\"$PATH_INFO\" onsubmit=\"return juge(this)\">\n";
		echo "<tr>\n";
		echo "<td colspan=\"2\">修改用户密码</td>\n";
		echo "</tr>\n";
		echo "<tr>\n";
		echo "<td>输入旧密码：</td>\n";
```

```
        echo "<td>";
        echo "<input type=\"password\" name=\"password\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入新密码：</td>\n";
        echo "<td>";
        echo "<input type=\"password\" name=\"newpassword\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>再输入一次：</td>\n";
        echo "<td>";
        echo "<input type=\"password\" name=\"repassword\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center>";
        echo "<input type=submit value=\"提交\">";
        echo "</center></td>\n";
        echo "</tr>\n";
        echo "</form>";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>\n";
        echo "</html>\n";
    }
    else
    {
        require "21-1.php";
        $password=md5($_POST[password]);
        $newpassword=md5($_POST[newpassword]);
        $id=$_COOKIE[id];
        $sql="select id from $table_user where name='$_COOKIE[user]' and password='$password'";
        $result=mysql_query($sql,$link);
        $nums=mysql_num_rows($result);
        if($nums<1)
        {
            echo "输入的用户密码错误！<p>";
            echo "请重新输入！";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=21-12.php\">";
        }
        else
        {
            $sql="update $table_user set password='$newpassword' where id='$id'";
            if(mysql_query($sql,$link))
            {
                echo "修改用户密码成功，现在返回首页";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=20-6.php\">";
            }
```

```
                    else
                    {
                        echo "修改用户密码失败，现在返回修改密码页";
                        echo "<meta http-equiv=\"refresh\" content=\"2; url=21-12.php\">";
                    }
                }
            }
        }
        ?>
```

保存上面代码为“21-12.php”，以备后用。

21.7.3　为图书添加新的分类

图书类别管理中的重要一项就是添加新的分类。在实际生活中，图书的种类是多种多样的，管理员应该为图书添加分类及二级分类，其实质是为分类表添加新的记录。

请参考以下代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>增加新的图书类别</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
require "21-1.php";
$sql="select admin from $table_user where id='$_COOKIE[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[0]!=3)
{
    echo "你没有权限执行这项操作！ ";
    exit();
}
else
{
    if(!$_POST[type])
    {
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><h2>创建图书分类第一步</h2></center></td>\n";
        echo "</tr>";
        echo "<tr>\n";
        echo "<td>选择创建类别</td>\n";
        echo "<td>";
        echo "<select size=\"1\" name=\"type\">\n";
```

```
            echo "<option value=\"1\">主类别</option>\n";
            echo "<option value=\"2\">分类别</option>\n";
            echo "</select>\n";
            echo "</td>\n";
            echo "</tr>";
            echo "<tr>\n";
            echo "<td colspan=\"2\"><center><input type=submit value=\"下一步\"></td>\n";
            echo "</tr>\n";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>";
            echo "</html>\n";

        }
        else if(!$_POST[type_name])
        {
            echo "<script language=\"javascript\">\n";
            echo "function juge(theForm)\n";
            echo "{\n";
            echo "\tif (theForm.type_name.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入类别名称！\");\n";
            echo "\t\ttheForm.topic_name.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
            echo "\tif (theForm.type_description.value == \"\")\n";
            echo "\t{\n";
            echo "\t\talert(\"请输入类别介绍！\");\n";
            echo "\t\ttheForm.topic_description.focus();\n";
            echo "\t\treturn (false);\n";
            echo "\t}\n";
            echo "}\n";
            echo "</script>\n";
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo    "<form    method=\"post\"    action=\"".$_SERVER[PHP_SELF]."\"         onsubmit=\"return
    juge(this)\">\n";
            echo "<tr>\n";
            echo "<td colspan=\"2\"><center><h2>创建图书分类第二步</h2></center></td>\n";
            echo "</tr>";
            echo "<input type=\"hidden\" name=\"type\" value=\"".$_POST[type]."\">";
            if($_POST[type]==2)
            {
                echo "<tr>\n";
                echo "<td>选择分类别所属主类</td>\n";
                echo "<td>";
                echo "<select size=\"1\" name=\"p_id\">\n";
                $sql="select id,type_name from $table_type where p_id=0";
                $result=mysql_query($sql,$link);
```

```
				while($rows=mysql_fetch_array($result))
				{
					echo "<option value=\"".$rows[id]."\">".$rows[type_name]."</option>\n";
				}
				echo "</select>\n";
				echo "</td>\n";
				echo "</tr>";
			}
			echo "<tr>\n";
			echo "<td>输入类别名称</td>\n";
			echo "<td>";
			echo "<input type=\"text\" name=\"type_name\">\n";
			echo "</td>\n";
			echo "</tr>";
			echo "<tr>\n";
			echo "<td>输入类别介绍</td>\n";
			echo "<td>";
			echo "<input type=\"text\" name=\"type_description\">\n";
			echo "</td>\n";
			echo "</tr>";
			echo "<tr>\n";
			echo "<td colspan=\"2\"><center><input type=button value=\" 上 一 步 \"
onclick=\"history.go(-1)\"><input type=submit value=\"下一步\"></td>\n";
			echo "</tr>\n";
			echo "</form>\n";
			echo "</table>\n";
			echo "</center>\n";
			echo "</body>";
			echo "</html>\n";
		}
		else
		{
			$type=$_POST[type];
			$type_name=$_POST[type_name];
			$type_description=$_POST[type_description];
			if($type==2)
			{
				$p_id=$_POST[p_id];
			}
			else $p_id=0;
			$sql="insert into
$table_type(p_id,type_name,type_description)values('$p_id','$type_name','$type_description')";
			if(mysql_query($sql,$link))
			{
				echo "增加新类别操作成功，现在返回首页！";
				echo "<meta http-equiv=\"refresh\" content=\"2; url=21-6\">";
			}
			else
			{
```

```
                    echo "增加新类别操作失败，现在返回！";
                    echo "<meta http-equiv=\"refresh\" content=\"2; url=21-13.php\">";
                }
        }
}
?>
```

将上述代码保存为“21-13.php”。当管理员调用该文件时，其执行结果如图 21.8 所示。从图 21.8 中选取相应的创建类别单击“下一步”按钮，执行结果如图 21.9 如示。

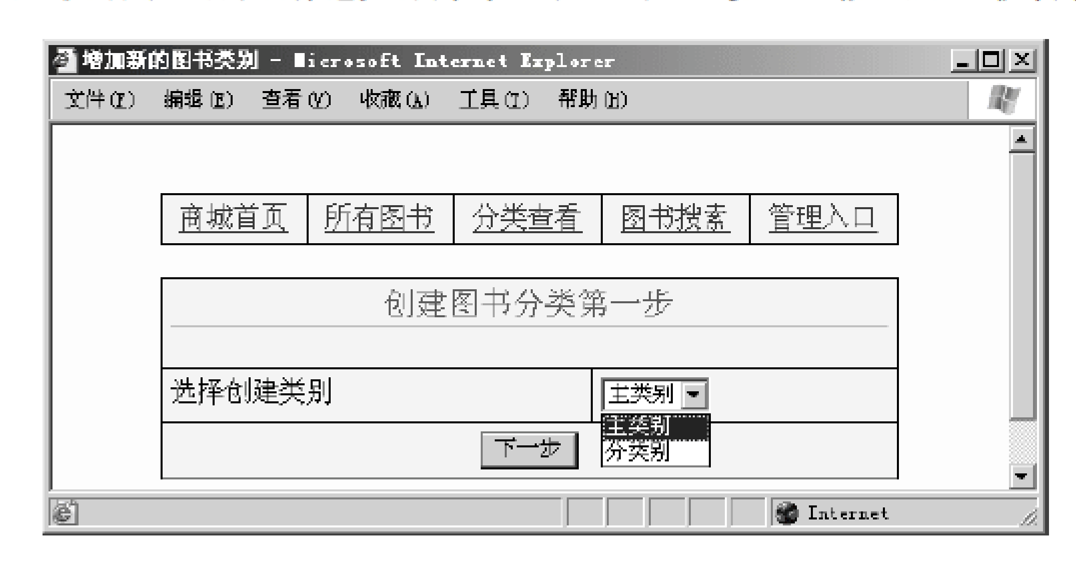


图 21.8　增加新类别的第 1 步



图 21.9　增加新类别的第 2 步

在图 21.8 中输入类别的名称及类别介绍，如果创建的类别是分类别，还要对该分类别所属主类别进行选择。输入完成后，就可以完成对相应类别的添加。

21.7.4　修改已经存在的分类

设置好的分类不一定是最合适的，所以也要允许管理员对分类进行修改，如修改分类名称、修改分类的说明、修改二级分类所属的主分类等。这些操作通过互动表单及强大的 MySQL 数据库语句都能够实现。

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改现有的图书类别</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
require "21-1.php";
$sql="select admin from $table_user where id='$_COOKIE[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[0]!=3)
{
        echo "你没有权限执行这项操作！";
        exit();
}
else
```

```
{
            if(!$_POST[id])
            {
            echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
            echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\">\n";
            echo "<tr>\n";
            echo "<td colspan=\"4\"><center><h2>修改图书分类第一步</h2></center></td>\n";
            echo "</tr>";
            echo "<tr>\n";
            echo "<td>选择图书分类</td>\n";
            echo "<td>分类类型</td>\n";
            echo "<td>图书分类名称</td>\n";
            echo "<td>该类别简介</td>\n";
            echo "</tr>";
            $sql="select id,p_id,type_name,type_description from $table_type";
            $result=mysql_query($sql,$link);
            while($rows=mysql_fetch_array($result))
            {
                  echo "<tr>\n";
                  echo "<td><input type=\"radio\" name=\"id\" value=\"".$rows[id]."\"></td>\n";
                  echo "<td>";
                  if($rows[p_id]==0)
                  {
                        echo "主分类";
                  }
                  else
                  {
                        echo "子分类";
                  }
                  echo "</td>\n";
                  echo "<td>".$rows[type_name]."</td>\n";
                  echo "<td>".$rows[type_description]."</td>\n";
                  echo "</tr>\n";
            }
            echo "<tr>\n";
            echo "<td colspan=\"4\"><center><input type=submit value=\"下一步\"></td>\n";
            echo "</tr>\n";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>";
            echo "</html>\n";

      }
      else if(!$_POST[type_name])
      {
            echo "<script language=\"javascript\">\n";
            echo "function juge(theForm)\n";
            echo "{\n";
```

```
        echo "\tif (theForm.type_name.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入类别名称！ \");\n";
        echo "\t\ttheForm.type_name.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "\tif (theForm.type_description.value == \"\")\n";
        echo "\t{\n";
        echo "\t\talert(\"请输入类别介绍！ \");\n";
        echo "\t\ttheForm.type_description.focus();\n";
        echo "\t\treturn (false);\n";
        echo "\t}\n";
        echo "}\n";
        echo "</script>\n";
        echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\" onsubmit=\"return
juge(this)\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><h2>修改图书分类第二步</h2></center></td>\n";
        echo "</tr>";
        echo "<input type=\"hidden\" name=\"id\" value=\"".$_POST[id]."\">";
        $sql="select * from $table_type where id='$_POST[id]'";
        $result=mysql_query($sql,$link);
        $rows=mysql_fetch_array($result);
        if($rows[p_id]!=0)
        {
            echo "<tr>\n";
            echo "<td>选择子类别所属主类</td>\n";
            echo "<td>";
            echo "<select size=\"1\" name=\"p_id\">\n";
            $sql2="select id,type_name from $table_type where p_id=0";
            $result2=mysql_query($sql2,$link);
            while($rows2=mysql_fetch_array($result2))
            {
                echo "<option value=\"".$rows2[id];
                if($rows2[id]==$rows[p_id]) echo " checked ";
                echo ">".$rows2[type_name]."</option>\n";
            }
            echo "</select>\n";
            echo "</td>\n";
            echo "</tr>";
        }
        echo "<tr>\n";
        echo "<td>输入类别名称</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"type_name\" value=\"".$rows[type_name]."\">\n";
        echo "</td>\n";
        echo "</tr>";
        echo "<tr>\n";
```

```
            echo "<td>输入类别介绍</td>\n";
            echo "<td>";
            echo "<input type=\"text\" name=\"type_description\" value=\"".$rows[type_description]."\">\n";
            echo "</td>\n";
            echo "</tr>";
            echo "<tr>\n";
            echo "<td colspan=\"2\"><center><input type=button value=\" 上 一 步 \"
onclick=\"history.go(-1)\"><input type=submit value=\"下一步\"></td>\n";
            echo "</tr>\n";
            echo "</form>\n";
            echo "</table>\n";
            echo "</center>\n";
            echo "</body>";
            echo "</html>\n";
        }
        else
        {
            $id=$_POST[id];
            $type=$_POST[type];
            $type_name=$_POST[type_name];
            $type_description=$_POST[type_description];
            if($_POST[p_id])
            {
                $p_id=$_POST[p_id];
            }
            else
            {
                $p_id=0;
            }
            $sql="update $table_type set
p_id='$p_id',type_name='$type_name',type_description='$type_description' where id=$id";
            if(mysql_query($sql,$link))
            {
                echo "修改图书分类操作成功，现在返回图书分类列表！";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=21-8\">";
            }
            else
            {
                echo "修改图书分类操作失败，现在返回！";
                echo "<meta http-equiv=\"refresh\" content=\"2; url=21-14.php\">";
            }
        }
    }
    ?>
```

将上面的代码保存为“21-14.php”，以备后用。管理员调用该页面时其执行结果如图 21.10 所示。该页面显示出了所有的图书类别。从中选取相应类别，单击前面的单选按钮，然后单击“下一步”按钮，执行结果如图 21.11 所示。



图 21.10　修改已有类别第 1 步执行结果

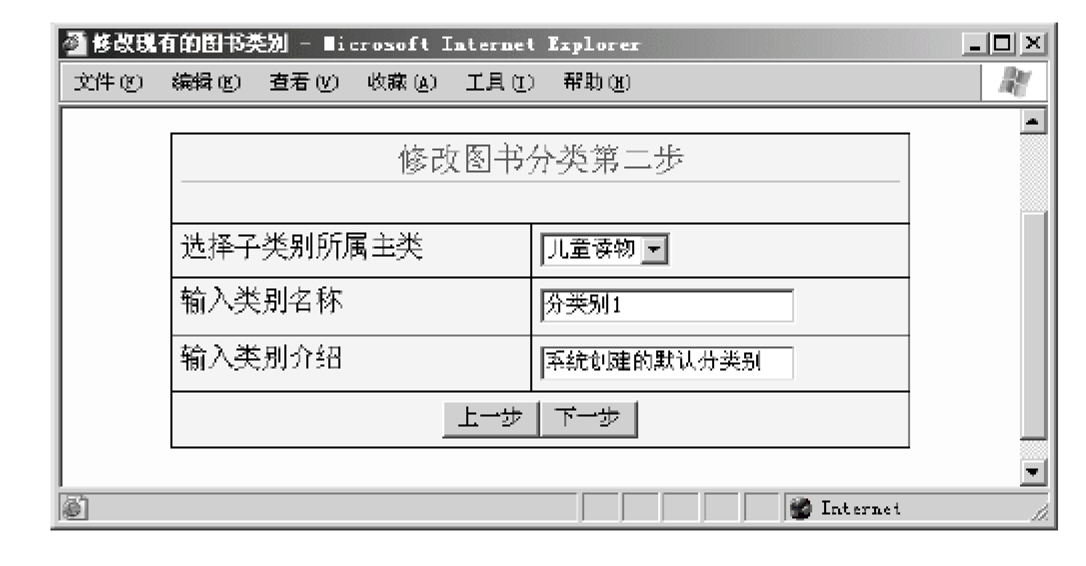


图 21.11　修改已有类别第 2 步执行结果

从图 21.11 可以看出，可以修改类别名称及类别介绍，如果选择修改的类别是子类别还可以重新选择该子类别所属的分类别。按需求对各项内容进行填写，单击“下一步”按钮，即可完成对本类别的修改。

21.7.5　增加新的图书

图书的库存量总是在不断地增加，所以增加新图书的功能是必不可少的。特别是在使用初期，必须把库存图书“入库”。这里说的入库并不是通常意义上的放入仓库，而是把所有图书的相关信息存入数据库的表中。使用下面代码即可实现这样的功能。

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>增加新的图书</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
require "21-1.php";
$sql="select admin from $table_user where id='$_COOKIE[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[0]!=3)
{
	echo "你没有权限执行这项操作！";
	exit();
}
else
{
	if(!$_POST[book_name])
	{
		$sql="select id,type_name from $table_type where p_id=0";
		$result=mysql_query($sql,$link);
		$i=0;
		while($rows=mysql_fetch_array($result))
		{
```

```
                    $j=0;
                    $temp[$i][0]=$rows[type_name];
                    $flag[$i][0]=$rows[id];
                    $sql2="select id,type_name from $table_type where p_id='$rows[id]'";
                    $result2=mysql_query($sql2,$link);
                    while($rows2=mysql_fetch_array($result2))
                    {
                        $j++;
                        $temp[$i][$j]=$rows2[type_name];
                        $flag[$i][$j]=$rows2[id];
                    }
                    $i++;
                }
                echo "<script language=javascript>
function Juge(theForm)
{
    if (theForm.book_name.value == \"\")
  {
    alert(\"请输入书名!\");
    theForm.book_name.focus();
    return (false);
  }
    if (theForm.book_author.value == \"\")
  {
    alert(\"请输入作者!\");
    theForm.book_author.focus();
    return (false);
  }
    if (theForm.cost.value == \"\")
  {
    alert(\"请输入书的价格!\");
    theForm.cost.focus();
    return (false);
  }
    if (theForm.book_num.value == \"\")
  {
    alert(\"请输入数量!\");
    theForm.book_num.focus();
    return (false);
  }
    if (theForm.book_description.value == \"\")
  {
    alert(\"请输入内容简介!\");
    theForm.book_description.focus();
    return (false);
  }
}
function change(){
for(var i=document.f.s_type.length;i>=0;i--) document.f.s_type.options[i]=null;
```

```
switch(document.f.m_type.options[document.f.m_type.selectedIndex].text){\n";
for($i=0;$i<count($temp);$i++)
{
    echo "case ".'\"".$temp[$i][0]."\":\n";
    for($j=1;$j<count($temp[$i]);$j++)
    echo                                    "document.f.s_type.options[".($j-1)."]=new
Option(\"".$temp[$i][$j]."\",\"".$flag[$i][$j]."\",false,false);\n";
    echo "break;\n";
}
echo "}\n}\n</script>\n";
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo    "<form    method=\"post\"    action=\"".$_SERVER[PHP_SELF]."\"    name=\"f\"
ENCTYPE=\"multipart/form-data\">\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><h2>增加新的图书第一步</h2></center></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入图书名称：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_name\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入图书作者：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_author\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入出版社：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_pub\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入售价：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_cost\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择所属类别：</td>\n";
        echo "<td>";
        echo "主类别：";
        echo "<select size=\"1\" name=\"m_type\"  onchange=\"change()\">\n";
        for($i=0;$i<count($temp);$i++)
        {
            echo "<option value=".$flag[$i][0].">".$temp[$i][0];
        }
```

```
        echo "</select>\n<br>";
        echo "分类别：";
        echo "<select size=\"1\" name=\"s_type\">\n";
        for($i=1;$i<count($temp[0]);$i++)
        {
            echo "<option value=".$flag[0][$i].">".$temp[0][$i];
        }
        echo "</select>\n";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入该书的数量：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_num\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入该书的简介：</td>\n";
        echo "<td>";
        echo "<textarea name=\"book_description\" cols=\"30\" rows=\"5\"></textarea>";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>上传该书的封面扫描图：</td>\n";
        echo "<td>";
        echo "<input type=\"file\" name=\"photo\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><input type=submit value=\"下一步\"></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
        echo "</table>\n";
        echo "</center>\n";
        echo "</body>";
        echo "</html>\n";
    }
    else
    {
        $book_name=$_POST[book_name];
        $book_author=$_POST[book_author];
        $book_pub=$_POST[book_pub];
        $book_cost=$_POST[book_cost];
        $m_type=$_POST[m_type];
        $s_type=$_POST[s_type];
        $book_num=$_POST[book_num];
        $book_description=$_POST[book_description];
        if($photo)
        {
            $filepath="uploads/";
```

```
            $file_temp=explode(".",$photo_name);
            $filename=$filepath.date(YmdHis).".".$file_temp[1];
            copy($photo,$filename);
            unlink($photo);
        }
        $sql="insert into $table_book
(book_name,book_author,book_pub,book_cost,book_type,book_num,book_description,book_photo)
values('$book_name','$book_author','$book_pub','$book_cost','$s_type','$book_num','$book_description','$filen
ame')";
        if(mysql_query($sql,$link))
        {
            $sql="update $table_type set type_num=type_num+1 where id='$m_type'";
            mysql_query($sql,$link);
            $sql="update $table_type set type_num=type_num+1 where id='$s_type'";
            mysql_query($sql,$link);
            echo "添加新的图书操作成功，现在返回查看全部图书页！";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=21-7\">";
        }
        else
        {
            echo "添加新的图书操作失败，现在返回重新输入！";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=21-15.php\">";
        }
    }
}
?>
```

把上述代码保存为“21-15.php”。这样在新书“入库”时就可以通过该文件来实现了。其执行结果如图 21.12 所示。



图 21.12　增加新的图书执行结果

在这里，可以输入图书的所有相关信息。包括图书的名称、作者、出版社、售价、所属主类别与分类别、存货数量、书的内容简介以及封面扫描图等。其中封面扫描图一项需要上传。全部填写完毕，单击“下一步”按钮，即可生成记录。

21.7.6　修改已有图书

图书在入库以后，由于种种原因，需要对图书的信息进行更改，这也是可以实现的。同更改用户注册信息一样，先给出一个人机交互界面，由管理员输入更改的内容，内容输入后，在后台对相应的内容进行更改。

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>修改已有的图书</title>\n";
echo "</head>\n";
echo "<body>\n";
echo "<center>\n";
require "21-3.php";
require "21-1.php";
$sql="select admin from $table_user where id='$_COOKIE[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[0]!=3)
{
    echo "你没有权限执行这项操作！ ";
    exit();
}
else
{
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    if(!$_POST[id])
    {
        $sql="select id from $table_book";                  //从列表中读出所有图书记录
        $result=mysql_query($sql,$link) or die(mysql_error());          //发送查找列表请求
        $num=mysql_num_rows($result);                       //获取结果条数
        $p_count=ceil($num/10);                             //总页数
        if ($_GET["page"]==0 && !$_GET["page"]) $page=1;//当前页
        else $page=$_GET["page"];
        if($num<1)                                          //如果没有记录
        {
            echo "<tr>\n";
            echo "<td>";
            echo "<center><h2>暂时还没有图书的记录</h2></center>";      //输出相应信息
            echo "</td>\n";
            echo "</tr>\n";
            exit();                                         //退出所有 PHP 代码
        }
        else                                                //如果有记录，则执行相应操作
        {
            $s=($page-1)*10;
            $sql="select * from $table_book order by id limit $s,10";
```

```
        $result=mysql_query($sql,$link);
        echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\">";
        echo "<tr>\n";
        echo "<td colspan=\"5\"><center><h2>已有图书第一步：选择记录</h2></center></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>选择</td>\n";
        echo "<td>书名</td>\n";
        echo "<td>作者</td>\n";
        echo "<td>价格</td>\n";
        echo "<td>类别</td>\n";
        while($rows=mysql_fetch_array($result))        //循环显示记录内容
        {
            echo "<tr>\n";
            echo "<td><input type=\"radio\" name=\"id\" value=\"".$rows[id]."\"></td>\n";
            echo "<td><a href=\"21-10.php?id=".$rows[id]."\">".$rows[book_name]."</a></td>\n";
                                                       //显示书名
            echo "<td>".$rows[book_author]."</td>\n";//显示作者
            echo "<td>".$rows[book_cost]."</td>\n";    //显示价格
            $sql2="select type_name from $table_type where id='$rows[book_type]'";
            $result2=mysql_query($sql2);
            $rows2=mysql_fetch_array($result2);
            echo "<td>".$rows2[0]."</td>";             //显示类别
            echo "</tr>\n";
        }
        echo "<tr>\n";
        echo "<td colspan=\"5\"><center><input type=\"submit\" value=\"修改选择项\"></center></td>\n";
        echo "</tr>\n";
        echo "</form>\n";
    }
    echo "</table>";
    $prev_page=$page-1;
    $next_page=$page+1;
    echo "              <p align=\"center\"> ";
    if ($page>1)
    {
        echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
    }
    if ($prev_page>=1)
    {
        echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
    }
    if ($next_page<=$p_count)
    {
        echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
    }
    if ($page<$p_count)
    {
        echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
    }
```

```
			echo "</center>";
			echo "</body>";
			echo "</html>";
		}
		else if(!$_POST[book_name])
		{
			$sql="select id,type_name from $table_type where p_id=0";
			$result=mysql_query($sql,$link);
			$i=0;
			while($rows=mysql_fetch_array($result))
			{
				$j=0;
				$temp[$i][0]=$rows[type_name];
				$flag[$i][0]=$rows[id];
				$sql2="select id,type_name from $table_type where p_id='$rows[id]'";
				$result2=mysql_query($sql2,$link);
				while($rows2=mysql_fetch_array($result2))
				{
					$j++;
					$temp[$i][$j]=$rows2[type_name];
					$flag[$i][$j]=$rows2[id];
				}
				$i++;
			}
					echo "<script language=javascript>
function Juge(theForm)
{
    if (theForm.book_name.value == \"\")
  {
    alert(\"请输入书名!\");
    theForm.book_name.focus();
    return (false);
  }
    if (theForm.book_author.value == \"\")
  {
    alert(\"请输入作者!\");
    theForm.book_author.focus();
    return (false);
  }
    if (theForm.cost.value == \"\")
  {
    alert(\"请输入书的价格!\");
    theForm.cost.focus();
    return (false);
  }
    if (theForm.book_num.value == \"\")
  {
    alert(\"请输入数量!\");
    theForm.book_num.focus();
    return (false);
```

```
    }
      if (theForm.book_description.value == \"\")
    {
      alert(\"请输入内容简介!\");
      theForm.book_description.focus();
      return (false);
    }
}
function change(){
for(var i=document.f.s_type.length;i>=0;i--) document.f.s_type.options[i]=null;
switch(document.f.m_type.options[document.f.m_type.selectedIndex].text){\n";
        for($i=0;$i<count($temp);$i++)
        {
            echo "case ".""\"".$temp[$i][0]."\":\n";
            for($j=1;$j<count($temp[$i]);$j++)
            echo                    "document.f.s_type.options[".($j-1)."]=new
Option(\"".$temp[$i][$j]."\",\"".$flag[$i][$j]."\",false,false);\n";
            echo "break;\n";
        }
        echo "}\n}\n</script>\n";
        $sql="select * from $table_book where id='$_POST[id]'";
        $result=mysql_query($sql,$link);
        $rows=mysql_fetch_array($result);
echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
        echo     "<form     method=\"post\"     action=\"".$_SERVER[PHP_SELF]."\"     name=\"f\"
ENCTYPE=\"multipart/form-data\">\n";
        echo "<input type=\"hidden\" name=\"id\" value=\"".$_POST[id]."\">";
        echo "<tr>\n";
        echo "<td colspan=\"2\"><center><h2>修改已有图书第二步：修改相关信息</h2></center></td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入图书名称：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_name\" value=\"".$rows[book_name]."\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入图书作者：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_author\" value=\"".$rows[book_author]."\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入出版社：</td>\n";
        echo "<td>";
        echo "<input type=\"text\" name=\"book_pub\" value=\"".$rows[book_pub]."\">";
        echo "</td>\n";
        echo "</tr>\n";
        echo "<tr>\n";
        echo "<td>输入售价：</td>\n";
```

```
    echo "<td>";
    echo "<input type=\"text\" name=\"book_cost\" value=\"".$rows[book_cost]."\">";
    echo "</td>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td>选择所属类别：</td>\n";
    echo "<td>";
    echo "主类别：";
    echo "<select size=\"1\" name=\"m_type\"   onchange=\"change()\">\n";
    for($i=0;$i<count($temp);$i++)
    {
        echo "<option value=".$flag[$i][0].">".$temp[$i][0];
    }
    echo "</select>\n<br>";
    echo "分类别：";
    echo "<select size=\"1\" name=\"s_type\">\n";
    for($i=1;$i<count($temp[0]);$i++)
    {
        echo "<option value=".$flag[0][$i].">".$temp[0][$i];
    }
    echo "</select>\n";
    echo "</td>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td>输入该书的数量：</td>\n";
    echo "<td>";
    echo "<input type=\"text\" name=\"book_num\" value=\"".$rows[book_num]."\">";
    echo "</td>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td>输入该书的内容简介：</td>\n";
    echo "<td>";
    echo "<textarea name=\"book_description\" cols=\"30\" rows=\"5\">".$rows[book_description]."</textarea>";
    echo "</td>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td>上传该书的封面扫描图：<br>（如果无改变请留空）</td>\n";
    echo "<td>";
    echo "<input type=\"file\" name=\"photo\">";
    echo "</td>\n";
    echo "</tr>\n";
    echo "<tr>\n";
    echo "<td colspan=\"2\"><center><input type=submit value=\"下一步\"></td>\n";
    echo "</tr>\n";
    echo "</form>\n";
    echo "</table>\n";
    echo "</center>\n";
    echo "</body>";
    echo "</html>\n";
}
```

```
	else
	{
		$id=$_POST[id];
		$book_name=$_POST[book_name];
		$book_author=$_POST[book_author];
		$book_pub=$_POST[book_pub];
		$book_cost=$_POST[book_cost];
		$m_type=$_POST[m_type];
		$s_type=$_POST[s_type];
		$book_num=$_POST[book_num];
		$book_description=$_POST[book_description];
		if($photo)
		{
			$filepath="uploads/";
			$file_temp=explode(".",$photo_name);
			$filename=$filepath.date(YmdHis).".".$file_temp[1];
			copy($photo,$filename);
			unlink($photo);
			$sql="update $table_book set
book_name='$book_name',book_author='$book_author',book_pub='$book_pub',book_cost='$book_cost',book
_type='$s_type',book_num='$book_num',book_description='$book_description',book_photo='$filename'";
		}
		else
		{
			$sql="update $table_book set
book_name='$book_name',book_author='$book_author',book_pub='$book_pub',book_cost='$book_cost',book
_type='$s_type',book_num='$book_num',book_description='$book_description'";
		}
		if(mysql_query($sql,$link))
		{
			echo "修改已经有的图书操作成功，现在返回查看全部图书页！ ";
			echo "<meta http-equiv=\"refresh\" content=\"2; url=21-7\">";
		}
		else
		{
			echo "修改已经有的图书失败，现在返回重新输入！ ";
			echo "<meta http-equiv=\"refresh\" content=\"2; url=21-16\">";
		}
	}
}
?>
```

把以上代码保存为“21-16.php”。从以上代码可见，要对已有图书进行修改共分 3 步。第 1 步，选择想要修改的图书。第 2 步，填写修改内容。这两步都需要人工干预。第 3 步为后台操作，用输入的内容来替换已经存在的内容，从而完成修改操作。其执行结果如图 21.13 所示。

图 21.13 执行结果为修改已有图书的第 1 步，先选择想要更改的图书。通过选中相应图书前的单选按钮来选择相应图书，然后单击最下方的“修改选择项”按钮，开始对具体内容进行修改。执行结果如图 21.14 所示。




图 21.13　修改图书页面执行结果第 1 步

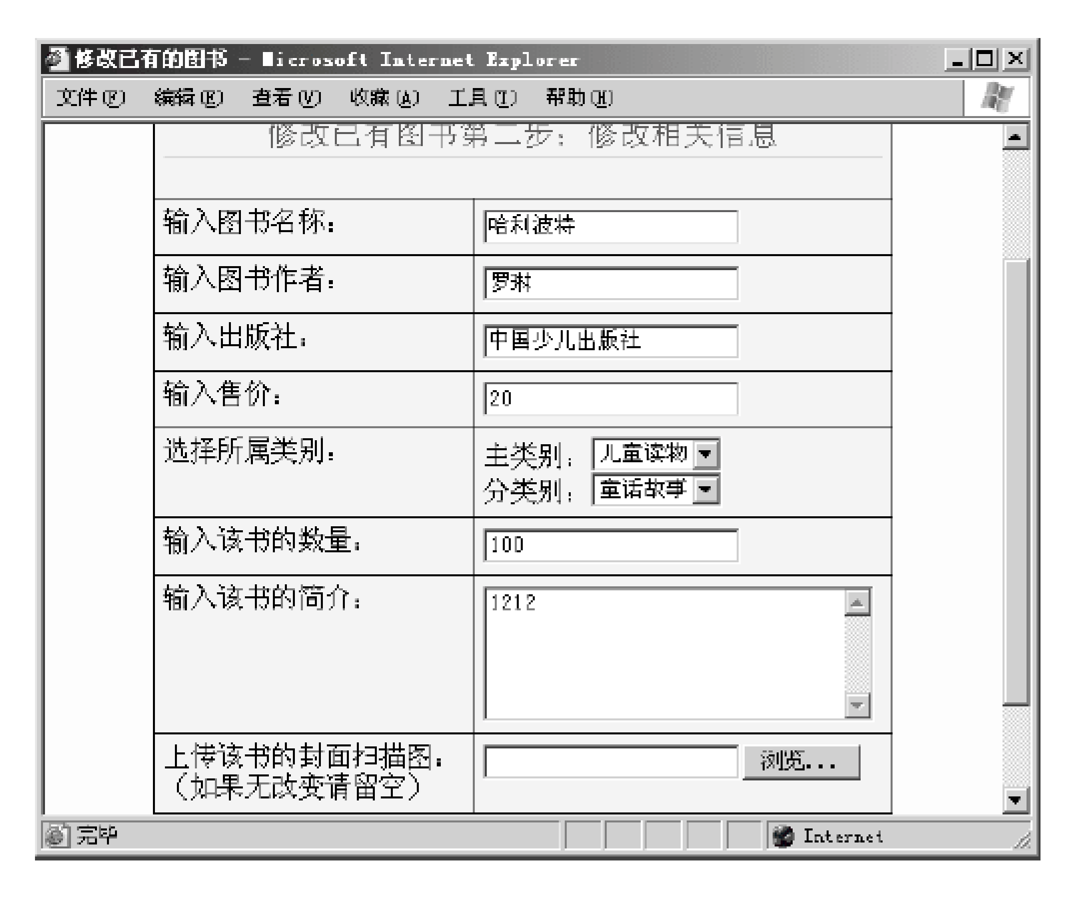


图 21.14　修改图书页面执行结果第 2 步

从图 21.14 可以看出，这里可以对图书的所有信息进行全方位的更改，具体过程与添加图书类似。输入完成后，单击最下方的“下一步”按钮，即可提交修改内容。

21.7.7　查看、处理所有未处理订单

用户的购买请求在提交后生成的订单，并不马上生效，需要管理员对其进行审核。只有在管理员对其进行处理后，才可能变为销售记录。所以，管理员可以查看并处理所有的用户订单。

下面给出管理员查看并处理所有用户订单的代码：

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>查看所有未处理订单</title>\n";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
require "21-1.php";
$sql="select admin from $table_user where id='$_COOKIE[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[0]!=3)
{
    echo "你没有权限执行这项操作！";
    exit();
}
else
{
    if(!$_POST[c])
    {
        echo "<script language=javascript>
        function checkall(form)
        {
```

```
                    for (var i=0;i<form.elements.length;i++)
                    {
                        var e = form.elements[i];
                        if (e.name != 'chkall')         e.checked = form.chkall.checked;
                    }
                }
</script>";
                echo "<h2>查看用户所有未处理的订单</h2>";
                echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
                echo "<form method=\"post\" action=\"".$_SERVER[PHP_SELF]."\">\n";
                echo "<tr>\n";
                echo "<td>处理</td>\n";
                echo "<td>提交人</td>\n";
                echo "<td>书号</td>\n";
                echo "<td>书数量</td>\n";
                echo "<td>订单总额</td>\n";
                echo "</tr>\n";
                $sql="select * from $table_order where order_state='false'";
                $result=mysql_query($sql,$link);
                while($rows=mysql_fetch_array($result))
                {
                    echo "<tr>\n";
                    echo "<td><input type=checkbox name=c[] value=".$rows[id]."></td>\n";
                    echo "<td>".$rows[order_user_name]."</td>\n";
                    echo          "<td><a              href=21-10.php?id=".$rows[order_book_id]."
target=\"_blank\">".$rows[order_book_id]."</a></td>\n";
                    echo "<td>".$rows[order_book_num]."</td>\n";
                    echo "<td>".$rows[order_cost]."</td>\n";
                    echo "</tr>\n";
                }
                echo "<tr>\n";
                echo "<td colspan=\"5\" align=\"center\">\n";
                echo "<input type=\"checkbox\" name=\"chkall\" value=\"on\" onclick=\"checkall(this.form)\">选择所有
记录";
                echo "<input type=\"submit\" value=\"提交所选记录\">";
                echo "</td>\n";
                echo "</tr>";
                echo "</form>";
                echo "</table>\n";
        }
        else
        {
                $c=$_POST[c];
                $time=date("Y 年 m 月 d 日");
                for($i=0;$i<count($c);$i++)
                {
                    $temp=$c[$i];
                    $sql="update $table_order set order_state='true' where id='$temp'";
                    mysql_query($sql,$link);
                    $sql2="insert into $table_sale (sale_order_id,sale_date)values('$temp','$time')";
```

```
                mysql_query($sql2,$link);
                $sql3="select order_book_id ,order_book_num from $table_order where id='$temp'";
                $result=mysql_query($sql3,$link);
                $rows=mysql_fetch_array($result);
                $sql4="update  $table_book  set  book_sale_num=book_sale_num+'$rows[1]'  ,  where
id='$rows[0]'";
                mysql_query($sql4,$link);
            }
            echo "处理订单成功,正在转到销售查看页";
            echo "<meta http-equiv=\"refresh\" content=\"2; url=21-18.php\">";

        }
}
?>
```

把以上代码保存为“21-17.php”，便于管理员处理未处理的订单。

21.7.8　销售记录的查看

管理员也可以查看所有的历史销售记录，从而了解所有商品的销售情况。

```
<?php
echo "<html>\n";
echo "<head>\n";
echo "<title>查看所有销售记录</title>\n";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
require "21-1.php";
$sql="select admin from $table_user where id='$_COOKIE[id]'";
$result=mysql_query($sql,$link);
$rows=mysql_fetch_array($result);
if($rows[0]!=3)
{
    echo "你没有权限执行这项操作！ ";
    exit();
}
else
{
    echo "<h2>查看所有的销售记录</h2>";
    echo "<table width=\"80%\" cellpadding=\"1\" cellspacing=\"1\">\n";
    $sql="select id from $table_sale";
    $result=mysql_query($sql,$link);
    $num=mysql_num_rows($result);
    $p_count=ceil($num/10);
    if ($_GET["page"]==0 && !$_GET["page"]) $page=1; //当前页
    else $page=$_GET["page"];
    if($num<1)                                          //如果没有记录
    {
```

```
        echo "<tr>\n";
        echo "<td>";
        echo "<center><h2>暂时还没有图书的记录</h2></center>";          //输出相应信息
        echo "</td>\n";
        echo "</tr>\n";
        exit();                                                         //退出所有 PHP 代码
    }
    else                                                                //如果有记录则执行相应操作
    {
        $s=($page-1)*10;
        $sql="select * from $table_sale order by id limit $s,10";
        $result=mysql_query($sql,$link);
        echo "<tr>\n";
        echo "<td>销售 ID</td>\n";
        echo "<td>提交人</td>\n";
        echo "<td>购书 ID</td>\n";
        echo "<td>购书价格</td>\n";
        echo "<td>处理日期</td>\n";
        echo "</tr>\n";
        while($rows=mysql_fetch_array($result))
        {
            echo "<tr>\n";
            echo "<td>".$rows[id]."</td>\n";
            $sql2="select * from $table_order where id='$rows[id]'";
            $result2=mysql_query($sql2,$link);
            $rows2=mysql_fetch_array($result2);
            echo "<td>".$rows2[order_user_name]."</td>\n";
            echo "<td>".$rows2[order_book_id]."</td>\n";
            echo "<td>".$rows2[order_cost]."</td>\n";
            echo "<td>".$rows[sale_date]."</td>\n";
            echo "</tr>\n";
        }
        echo "</table>";
        $prev_page=$page-1;
        $next_page=$page+1;
        echo "              <p align=\"center\"> ";
        if ($page>1)
        {
            echo "<a href='$PATH_INFO?page=1'>第一页</a> | ";
        }
        if ($prev_page>=1)
        {
            echo "<a href='$PATH_INFO?page=$prev_page'>上一页</a> | ";
        }
        if ($next_page<=$p_count)
        {
            echo "<a href='$PATH_INFO?page=$next_page'>下一页</a> | ";
        }
        if ($page<$p_count)
        {
```

```
            echo "<a href='$PATH_INFO?page=$p_count'>最后一页</a></p>\n";
        }
        echo "</center>";
        echo "</body>";
        echo "</html>";
    }
}
?>
```

保存以上代码为“21-18.php”，可以便于管理员查看所有销售记录。

21.7.9　当前登录用户退出页面

登录用户要求可以退出系统。实现方式很简单，就是清空当前用户注册的 Cookie 值，并清空所有的购买商品记录。

下面给出具体实现代码：

```
<?php
$temp=array_keys($_COOKIE);
for($i=0;$i<count($temp);$i++)                    //遍历 COOKIE 数组
{
    setcookie("$temp[$i]","");                    //清空所有 COOKIE 记录
}
echo "<html>\n";
echo "<head>\n";
echo "<title>退出登录成功</title>\n";
echo "<meta http-equiv=\"refresh\" content=\"2; url=21-6.php\">";
echo "</head>\n";
echo "<body>\n";
require "21-3.php";
echo "退出登录成功，现在转到首页";
echo "</body>";
echo "</html>";
```

把以上代码保存为“21-19.php”。当用户需要退出系统时，调用该页面，即可清空所有通过该系统注册的 Cookie 值，从而完成退出动作。

以上全部页面创建完成。该系统的创建过程告一段落。

21.8　本 章 小 结

这一章介绍了如何使用 PHP 和 MySQL 来构建一个简单的网上商城全站系统。详细地介绍了一个网上商城所应具备的基本功能，及如何用 PHP 来实现这些功能。通过本章的学习，读者对如何用 PHP 构建网上商城会有一个深刻的认识。本章所创建的网上商城功能有限，学完本书的读者完全有能力来进一步丰富它的功能，使它更加完善。